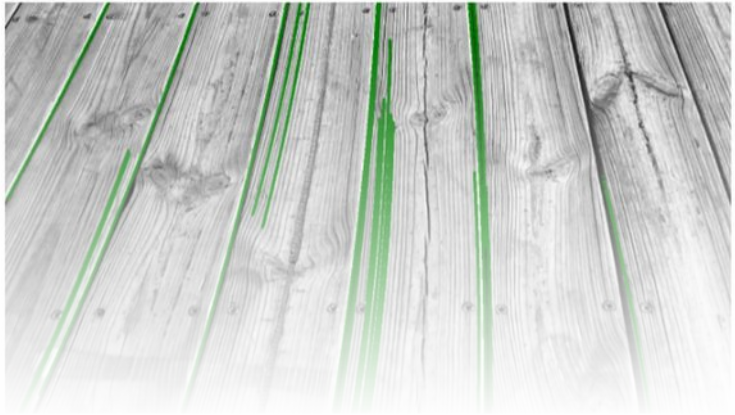



3D Reconstruction from Rolling Shutter Cameras

Theory and Practice

Petr Hrubý

petrh@kth.se

PETR HRUBÝ

- Working on minimal solvers for camera pose estimation and theoretical understanding thereof

- **CV:**
  - Postdoc at KTH (2025-Present)
  - PhD: ETH Zürich (2021-2025)
  - BSc. + MSc.: CTU in Prague, (2016-2021)

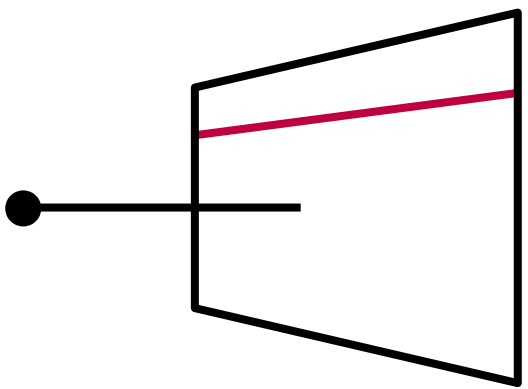
Content

- SfM and Rolling shutter cameras
- Projections of points and lines
- Relative pose from RS images

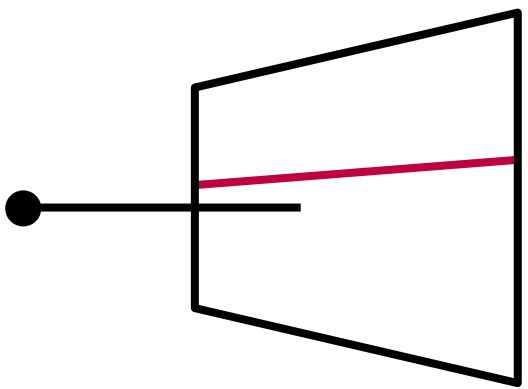
Structure from Motion (SfM)

Images of the Sceaux Castle by Pierre Moulon.

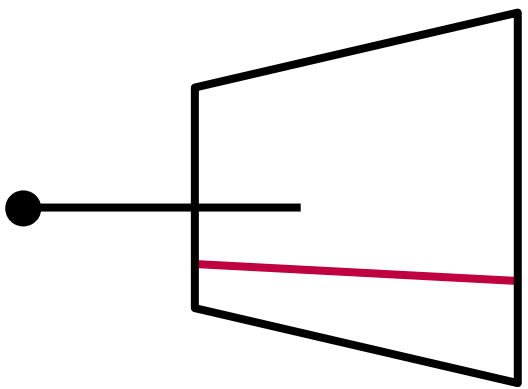
Rolling Shutter Camera



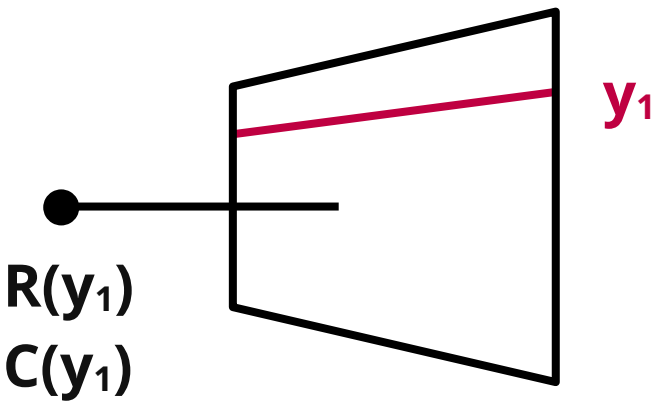
Rolling Shutter Camera



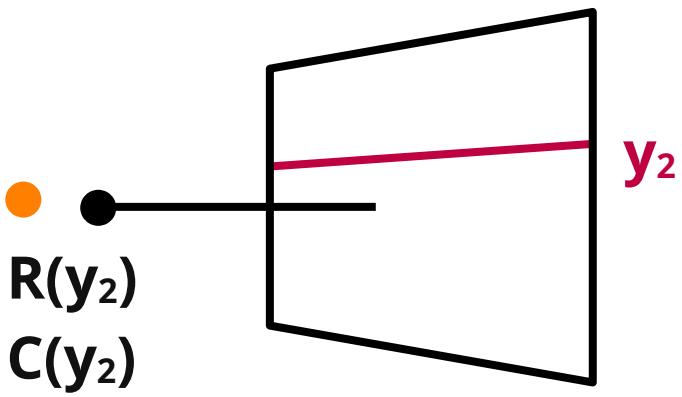
Rolling Shutter Camera



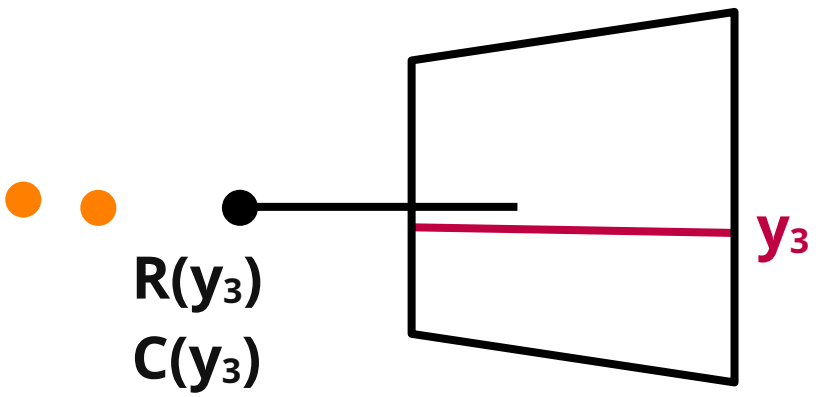
Rolling Shutter Camera



Rolling Shutter Camera



Rolling Shutter Camera



Motivation

- Rolling shutter cameras are cheap and they are everywhere
- Usually resolved by assuming global shutter or by rectification (removing RS effect)

Motivation

- Rolling shutter cameras are cheap and they are everywhere
- ~~Usually resolved by assuming global shutter or by rectification (removing RS effect)~~
- **Leveraging RS effect has benefits for 3D reconstruction!!!**

Benefits

- GS cameras require at least 2 images to do 3D reconstruction.

Benefits

- GS cameras require at least 2 images to do 3D reconstruction.
- With RS cameras, we only need one.

Single-View Rolling-Shutter SfM

Sofía Errázuriz Muñoz[1], Kim Kiehn[1], Petr Hruby[1], and Kathlén Kohn[1,2]

[1] KTH Royal Institute of Technology, Sweden
[2] Digital Futures, Sweden
sofiaerr@kth.se

Abstract. Rolling-shutter (RS) cameras are ubiquitous, but RS SfM (structure-from-motion) has not been fully solved yet. This work suggests an approach to remedy this: We characterize RS single-view geometry of observed world points or lines. Exploiting this geometry, we describe which motion and scene parameters can be recovered from a single RS image and systematically derive minimal reconstruction problems. We evaluate several representative cases with proof-of-concept solvers, highlighting both feasibility and practical limitations.

Keywords: rolling-shutter geometry · camera pose · minimal problems

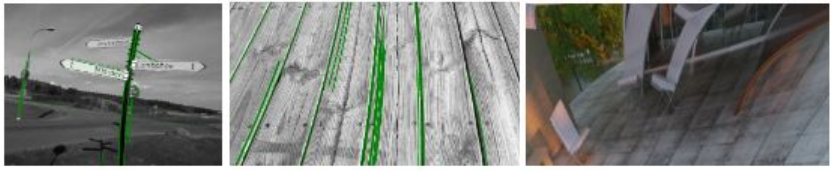


Fig. 1: Examples of RS images: *left:* iPhone 3GS sequence from [19], *middle*: sequence 06 from [26], *right:* synthetic image generated by [50].

Benefits

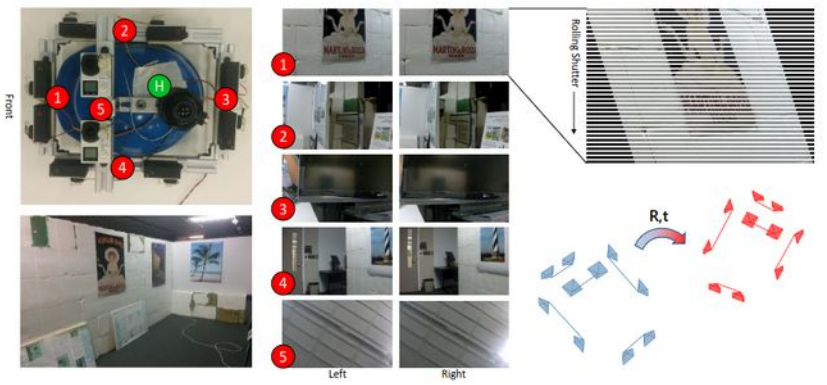
- GS cameras require at least 2 images to do 3D reconstruction.
- With RS cameras, we only need one.
- And they have much higher framerate

Rolling Shutter and Radial Distortion are Features for High Frame Rate Multi-camera Tracking

Akash Bapat, True Price, Jan-Michael Frahm
Department of Computer Science, The University of North Carolina at Chapel Hill
{akash, jtprice, jmf}@cs.unc.edu

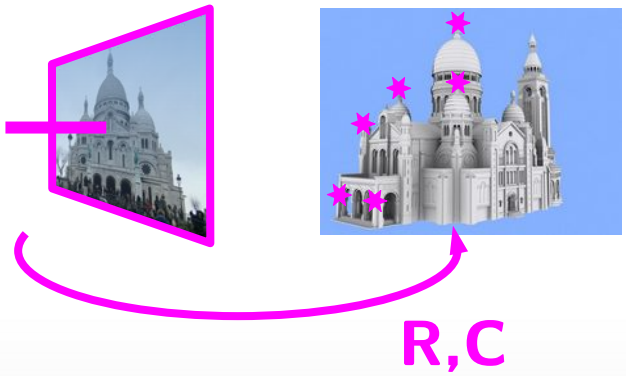
Towards Kilo-Hertz 6-DoF Visual Tracking Using an Egocentric Cluster of Rolling Shutter Cameras

Akash Bapat, Enrique Dunn, *Member, IEEE*, and Jan-Michael Frahm, *Member, IEEE*



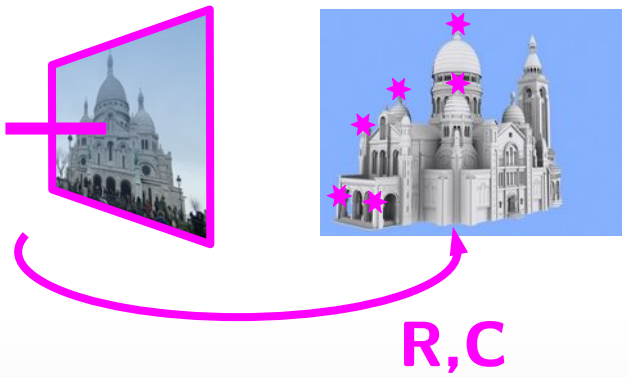
Global Shutter Pose vs. Rolling Shutter Pose

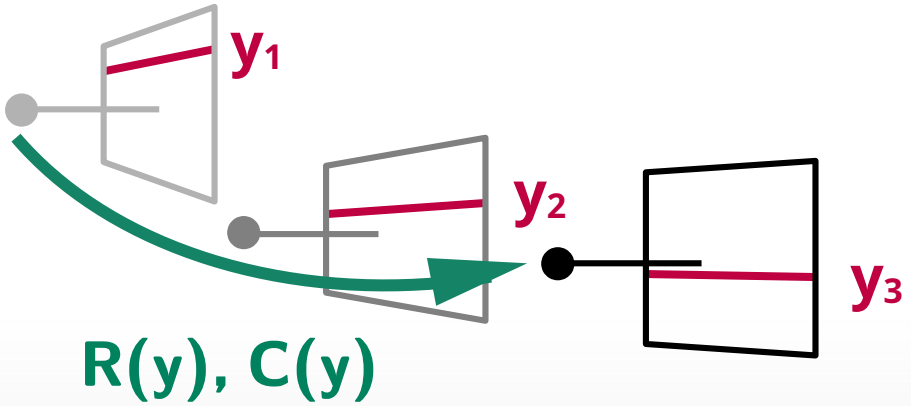
Global Shutter Pose vs. Rolling Shutter Pose



- **Single pose** per frame
- Simple modelling
  - Single **R**, **C**

Global Shutter Pose vs. Rolling Shutter Pose



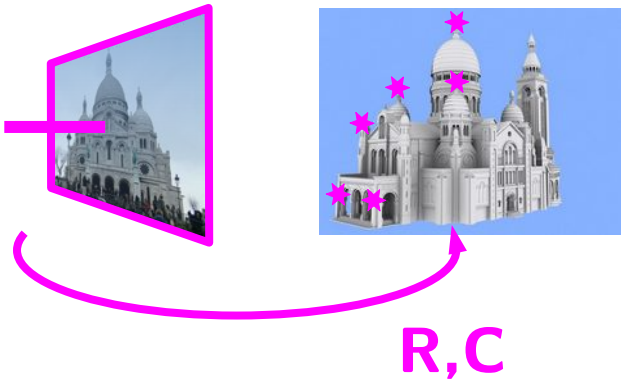


- **Single pose** per frame
- Simple modelling
  - Single **R**, **C**

- Need to model a **trajectory**
- Function **y → R(y), C(y)**
- Also for event cameras

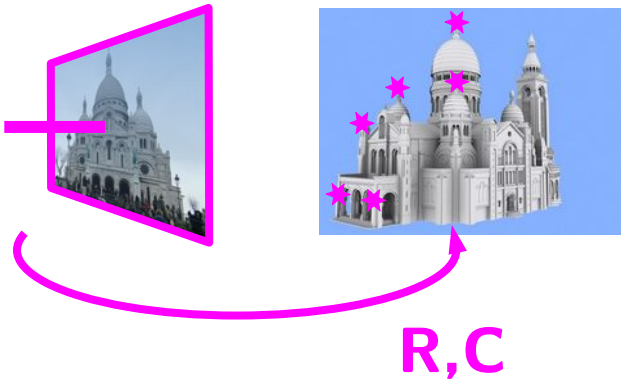
Modelling Global Shutter Pose

- Rotation + Camera Center



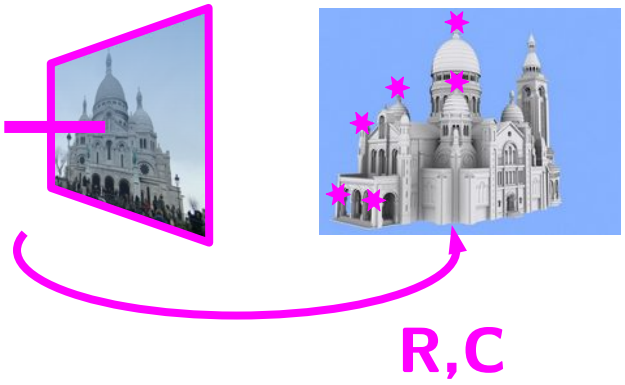
Modelling Global Shutter Pose

- Rotation + Camera Center
- **Center**: single vector $\mathbf{C} \in \mathbb{R}^3$



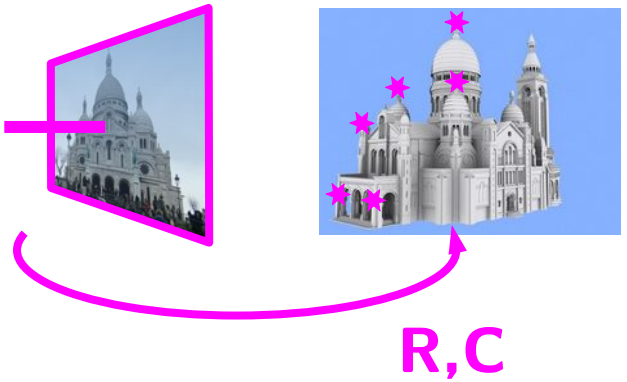
Modelling Global Shutter Pose

- Rotation + Camera Center
- **Center**: single vector $\mathrm{C} \in \mathbb{R}^3$
- **Rotation**:
  - Rotation Matrix: $\mathrm{R} \in \mathbb{R}^{3,3}$, $\mathrm{R}^{\mathrm{T}}\mathrm{R} = \mathrm{I}$, $\det(\mathrm{R}) = 1$
  - Quaternion: $\mathrm{q} \in \mathbb{R}^4$, $||\mathrm{q}|| = 1$
  - Cayley vector $\mathrm{A} \in \mathbb{R}^3$, $\mathrm{R} = \mathrm{C2R}(\mathrm{A})$
  - Angle + axis (not polynomial!)



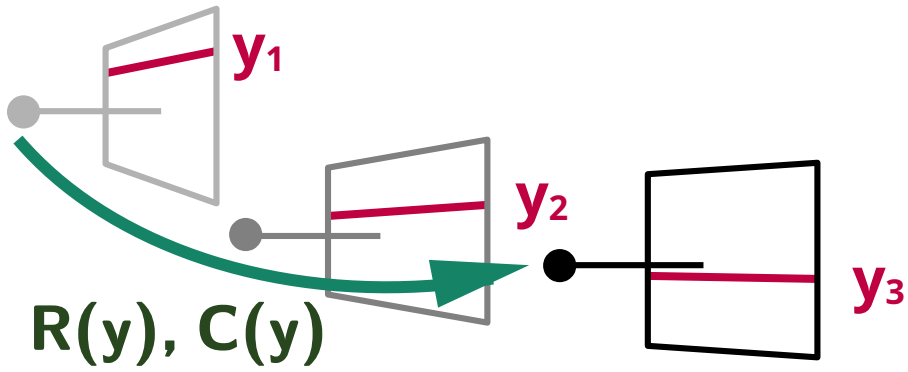
Modelling Global Shutter Pose

- Rotation + Camera Center
- **Center**: single vector $\mathrm{C} \in \mathbb{R}^3$
- **Rotation**:
  - Rotation Matrix: $\mathrm{R} \in \mathbb{R}^{3,3}$, $\mathrm{R}^{\mathrm{T}}\mathrm{R} = \mathrm{I}$, $\det(\mathrm{R}) = 1$
  - Quaternion: $\mathrm{q} \in \mathbb{R}^4$, $||\mathrm{q}|| = 1$
  - Cayley vector $\mathrm{A} \in \mathbb{R}^3$, $\mathrm{R} = \mathrm{C2R}(\mathrm{A})$
  - Angle + axis (not polynomial!)
- **All these model the same rotations***.



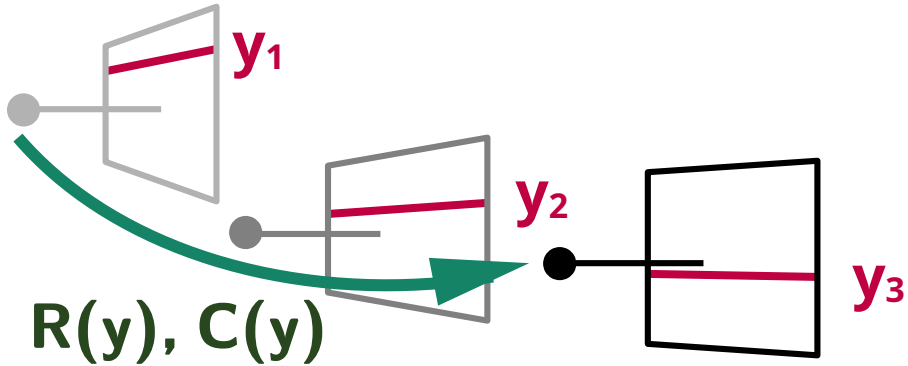
Modelling Rolling Shutter Pose

- Need to model a trajectory.
- Function: $y \rightarrow R(y), C(y)$



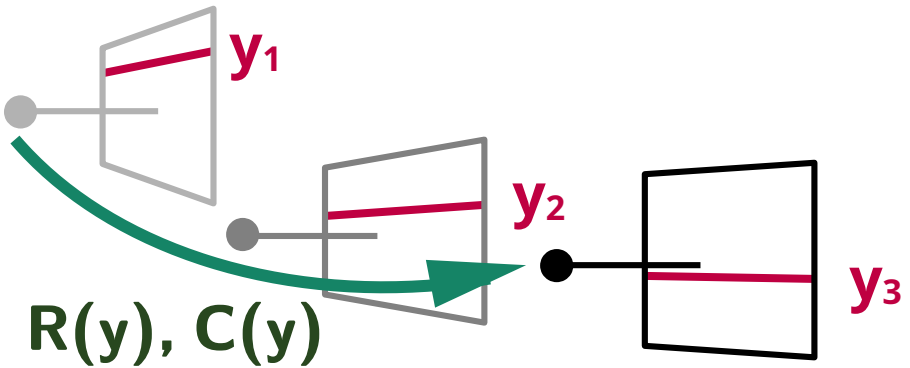
Modelling Rolling Shutter Pose

- Need to model a trajectory.
- Function: $y \rightarrow R(y), C(y)$
- Camera center C(y):
  - **Constant motion**: $C(y) = C_0$
  - **Constant velocity**: $C(y) = C_0 + y \cdot V$
  - **Accelerated motion**: $C(y) = C_0 + y \cdot V + y^2 \cdot W$
  - **Splines**: ensure smooth trajectory, used for SLAM



Modelling Rolling Shutter Pose

- Need to model a trajectory.
- Function: **$y \rightarrow R(y), C(y)$**
- Camera center C(y):
  - **Constant motion**: $C(y) = C_0$
  - **Constant velocity**: $C(y) = C_0 + y \cdot V$
  - **Accelerated motion**: $C(y) = C_0 + y \cdot V + y^2 \cdot W$
  - **Splines**: ensure smooth trajectory, used for SLAM
- **Each of these models different trajectories!**

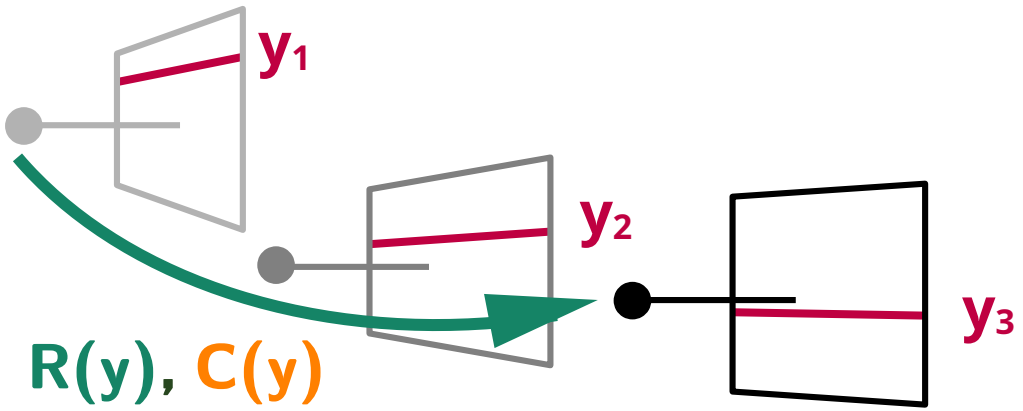


Modelling Rolling Shutter Rotation

- **Constant angular velocity**
  - $\mathrm{R}(\mathrm{y}) = (\mathrm{I} + \sin(\mathrm{y}\cdot\omega)[\mathrm{A}]_\mathrm{x} + (1\text{-}\cos(\mathrm{y}\cdot\omega))[\mathrm{A}]_\mathrm{x}^2) \cdot \mathrm{R}_0$
  - not polynomial
- **Cayley parameterization**
  - $\mathrm{R}(\mathrm{y}) = \mathrm{C2A}(\mathrm{y}\cdot\mathrm{A}_1) \cdot \mathrm{R}_0$:
- **Linearized motion**: $\mathrm{R}(\mathrm{y}) = (\mathrm{I} + [\mathrm{A}_1]_\mathrm{x}) \cdot \mathrm{R}_0$
  - Taylor appx. of both previous models, $\mathrm{R}^\mathrm{T}\mathrm{R} \neq \mathrm{I}$!
  - Lower degree, often used in minimal solvers
- **Splines**: ensure smooth trajectory, used for SLAM

Modelling Rolling Shutter Trajectory

- Camera projection for scanline y: linear map $\mathbb{P}^3 \rightarrow \mathbb{P}^2$
  - Represented with matrix **R(y) [I | -C(y)]**
  - Rotation **R(y)**: $\mathbb{R} \rightarrow$ SO(3), Camera center **C(y)**:$\mathbb{R} \rightarrow \mathbb{R}^3$

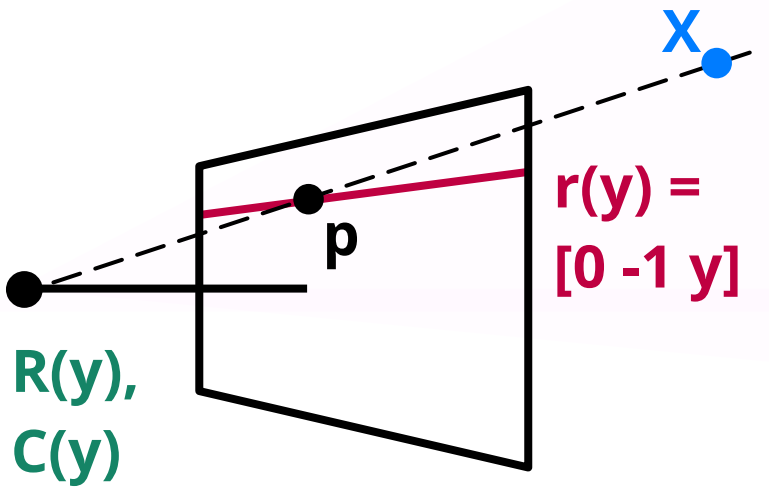


Modelling Rolling Shutter Trajectory

- Camera projection for scanline y: linear map $\mathbb{P}^3 \rightarrow \mathbb{P}^2$
  - Represented with matrix **R(y) [I | -C(y)]**
  - Rotation **R(y)**: $\mathbb{R} \rightarrow$ SO(3), Camera center **C(y)**: $\mathbb{R} \rightarrow \mathbb{R}^3$
- **R(y)** = $\frac{1}{1+\alpha^2+\beta^2+\gamma^2}\begin{bmatrix} 1+\alpha^2-\beta^2-\gamma^2 & 2(\alpha\beta-\gamma) & 2(\alpha\gamma+\beta) \\ 2(\alpha\beta+\gamma) & 1-\alpha^2+\beta^2-\gamma^2 & 2(\beta\gamma-\alpha) \\ 2(\alpha\gamma-\beta) & 2(\beta\gamma+\alpha) & 1-\alpha^2-\beta^2+\gamma^2 \end{bmatrix}$.
  - α,β,γ: polynomials of degree **δ**
- **C(y)** = $[a\ b\ c]^T$
  - a,b,c: polynomials of degree **d**

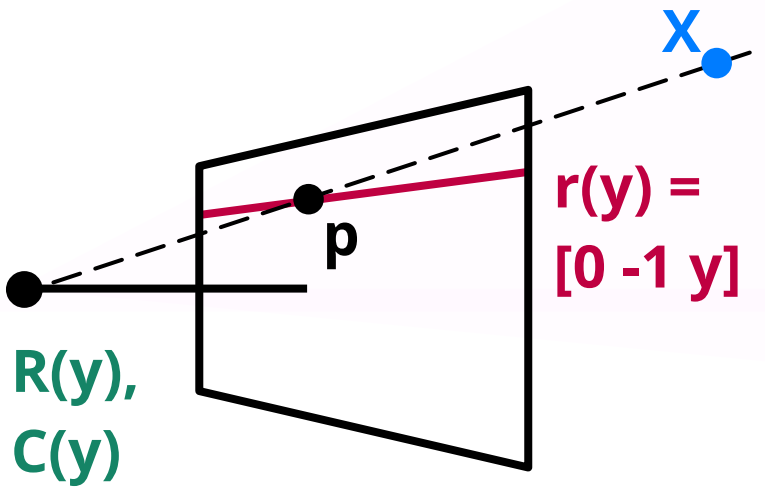
Projecting Points

- If $\mathbf{X}$ projects onto scanline $\mathbf{r(y)}$, $\mathbf{R(y)}\cdot(\mathbf{X}\text{-}\mathbf{C(y)})$ lies on $\mathbf{r(y)}$



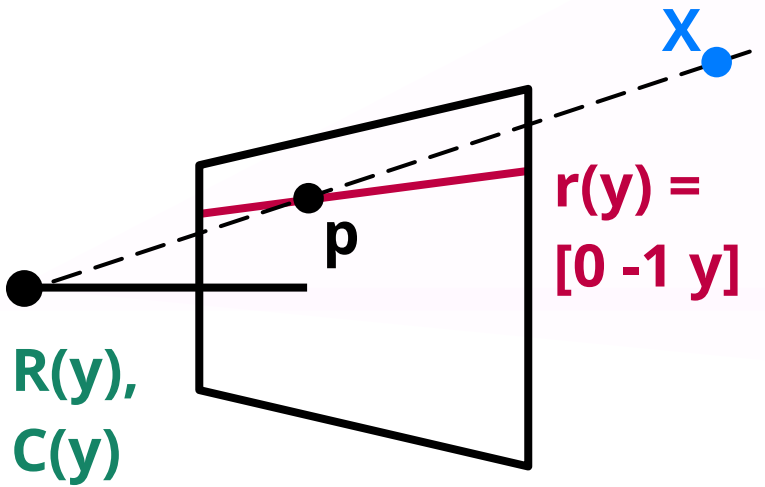
Projecting Points

- If $\mathbf{X}$ projects onto scanline $\mathbf{r(y)}$, $\mathbf{R(y)}\cdot(\mathbf{X}\text{-}\mathbf{C(y)})$ lies on $\mathbf{r(y)}$
- $\mathbf{r(y)}^{\mathrm{T}}\mathbf{R(y)}(\mathbf{X}\text{-}\mathbf{C(y)})=0$



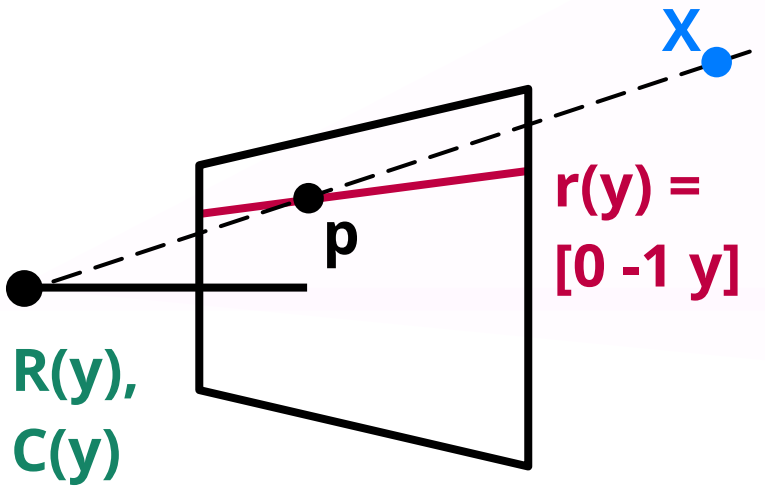
Projecting Points

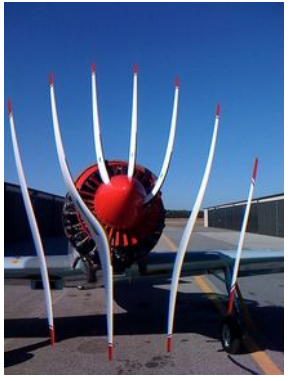
- If $\mathbf{X}$ projects onto scanline $\mathbf{r(y)}$,
  $\mathbf{R(y)}\cdot(\mathbf{X}\text{-}\mathbf{C(y)})$ lies on $\mathbf{r(y)}$
- $\mathbf{r(y)}^{\mathrm{T}}\mathbf{R(y)}(\mathbf{X}\text{-}\mathbf{C(y)})=0$
  - a nonlinear equation
  - multiple solutions in $\mathbf{y}$



Projecting Points

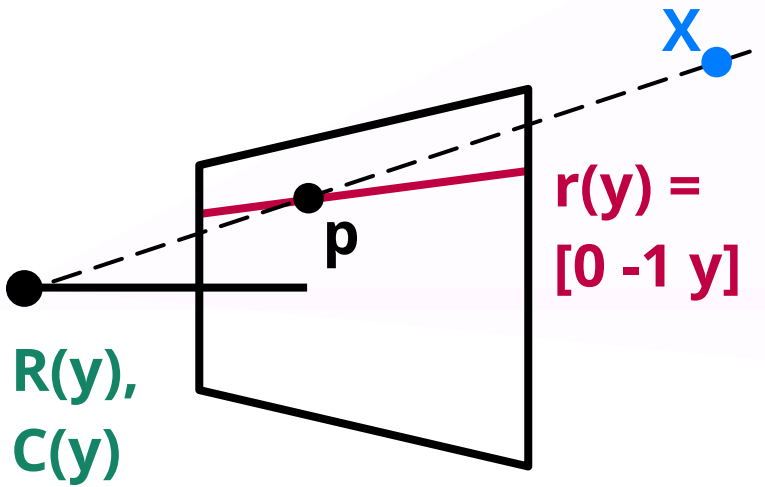
- If $\mathbf{X}$ projects onto scanline $\mathbf{r(y)}$, $\mathbf{R(y)}\cdot(\mathbf{X}\text{-}\mathbf{C(y)})$ lies on $\mathbf{r(y)}$
- $\mathbf{r(y)}^{\mathrm{T}}\mathbf{R(y)}(\mathbf{X}\text{-}\mathbf{C(y)})=0$
  - a nonlinear equation
  - multiple solutions in $\mathbf{y}$



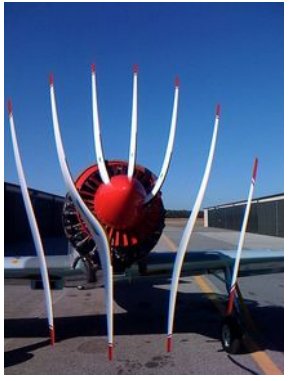


Projecting Points

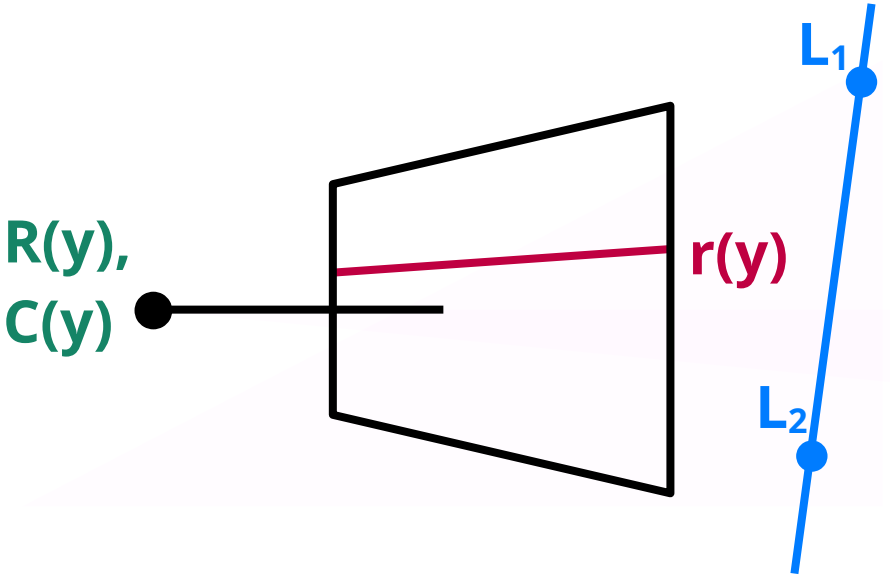
- If **X** projects onto scanline $\mathbf{r(y)}$,
  $\mathbf{R(y)}\cdot(\mathbf{X}\text{-}\mathbf{C(y)})$ lies on $\mathbf{r(y)}$



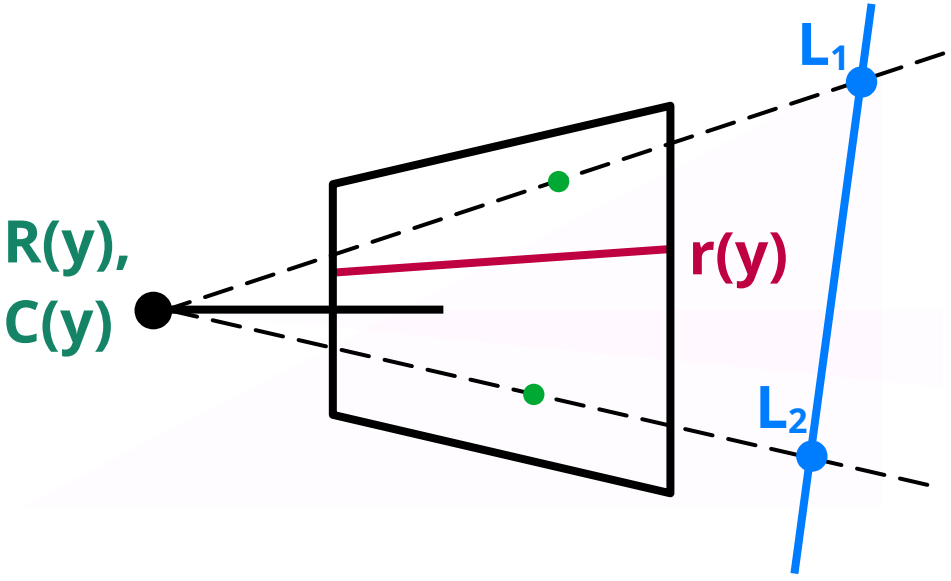
- $\mathbf{r(y)}^{\mathrm{T}}\mathbf{R(y)}(\mathbf{X}\text{-}\mathbf{C(y)})=0$
  - a nonlinear equation
  - multiple solutions in **y**
- **X** projects 1+d+2δ times
  - **C(y)** degree d polynomial, **R(y)**=C2R(A(y)), A(y) degree δ polynomial



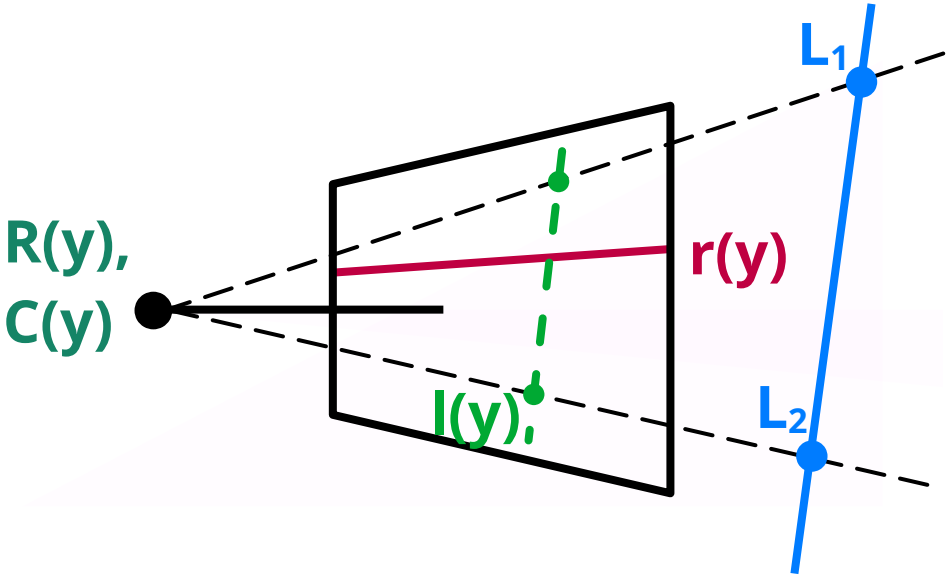
Projecting Lines



Projecting Lines

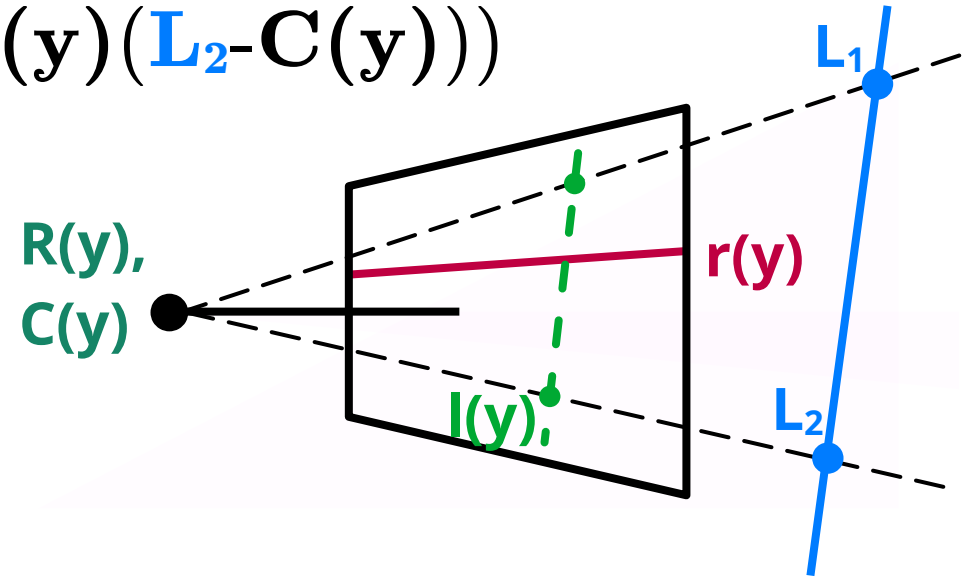


Projecting Lines



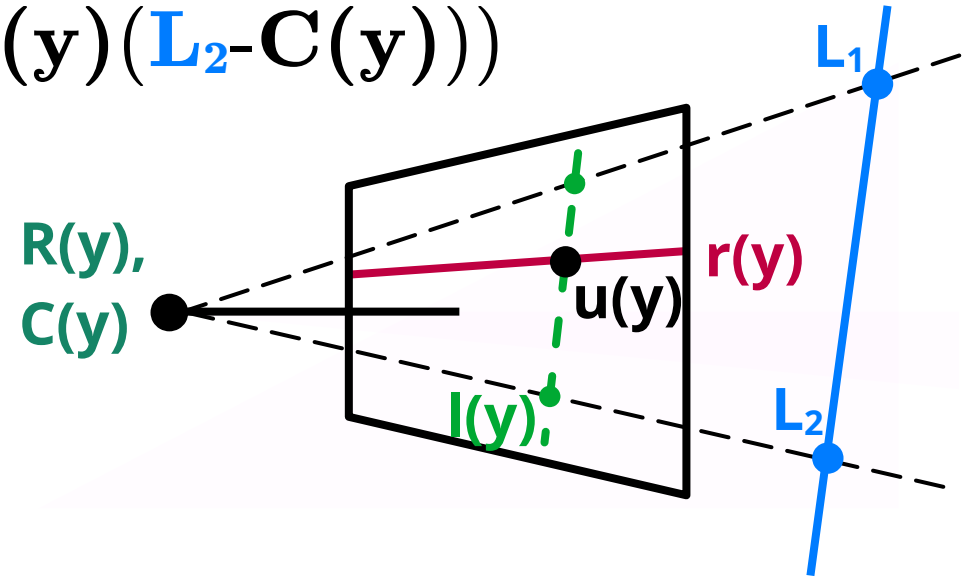
Projecting Lines

- $\mathbf{l(y)} = (\mathbf{R(y)}(\mathbf{L_1}\text{-}\mathbf{C(y)})) \times (\mathbf{R(y)}(\mathbf{L_2}\text{-}\mathbf{C(y)}))$



Projecting Lines

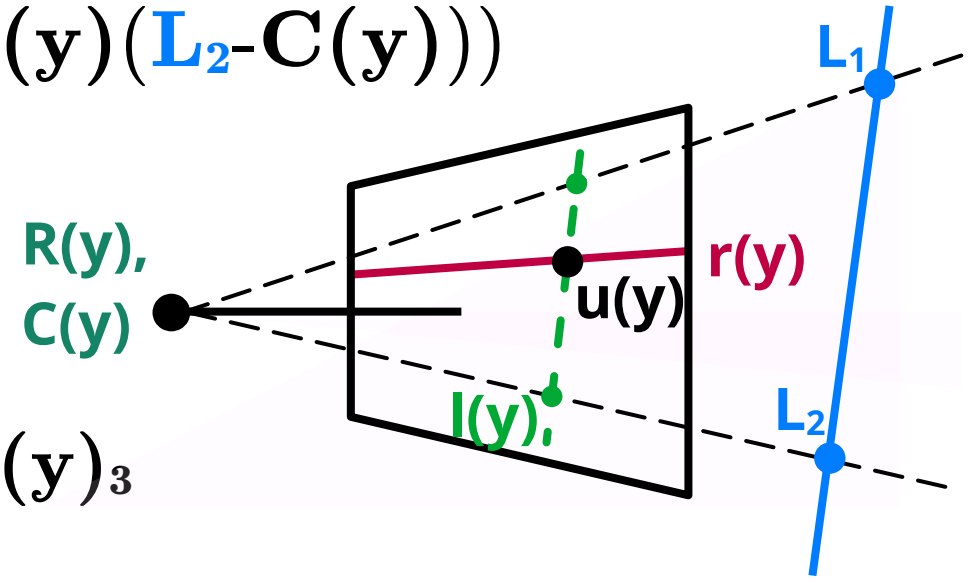
- $\mathbf{l(y)} = (\mathbf{R(y)}(\mathbf{L_1}\text{-}\mathbf{C(y)})) \times (\mathbf{R(y)}(\mathbf{L_2}\text{-}\mathbf{C(y)}))$
- $\mathbf{u(y)} = \mathbf{r(y)} \times \mathbf{l(y)}$



Projecting Lines

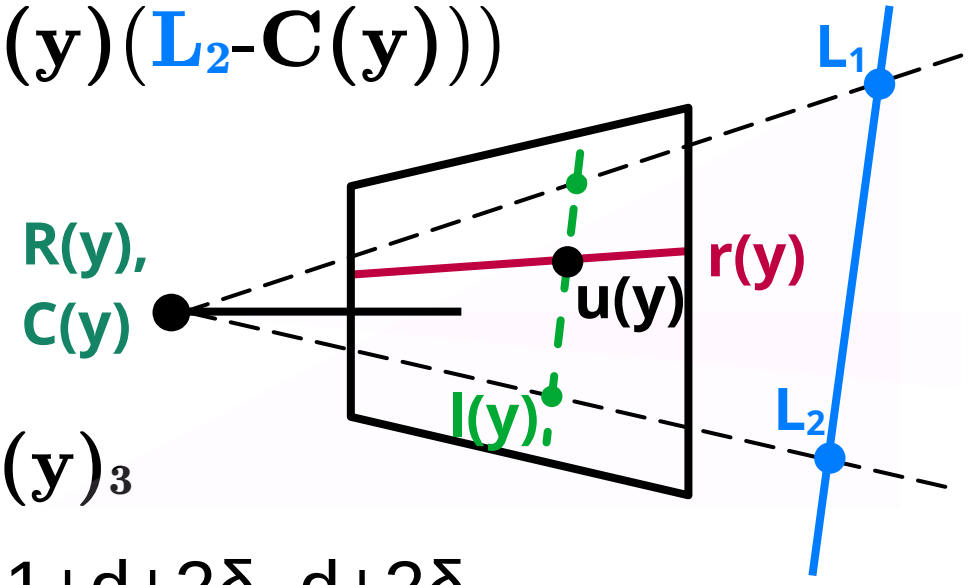
- $\mathbf{l(y)} = (\mathbf{R(y)}(\mathbf{L_1}\text{-}\mathbf{C(y)})) \times (\mathbf{R(y)}(\mathbf{L_2}\text{-}\mathbf{C(y)}))$

- $\mathbf{u(y)} = \mathbf{r(y)} \times \mathbf{l(y)}$
  - x coordinate of $\mathbf{u(y)}$: $\mathbf{u(y)_1}/\mathbf{u(y)_3}$



Projecting Lines

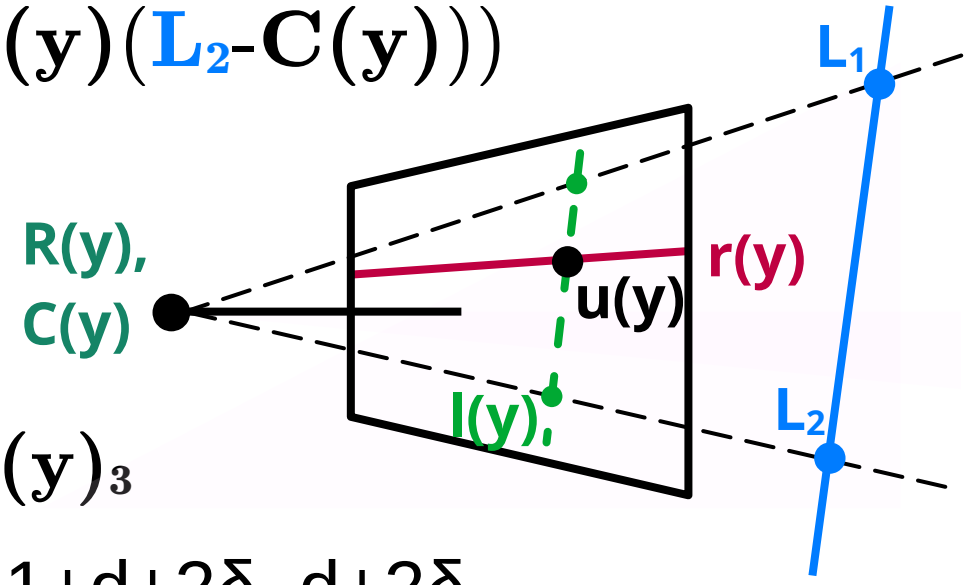
- $\mathbf{l(y)} = (\mathbf{R(y)}(\mathbf{L_1}\text{-}\mathbf{C(y)})) \times (\mathbf{R(y)}(\mathbf{L_2}\text{-}\mathbf{C(y)}))$



- $\mathbf{u(y)} = \mathbf{r(y)} \times \mathbf{l(y)}$
  - x coordinate of $\mathbf{u(y)}$: $\mathbf{u(y)_1}/\mathbf{u(y)_3}$
  - *rational function* of y, degrees 1+d+2δ, d+2δ

Projecting Lines

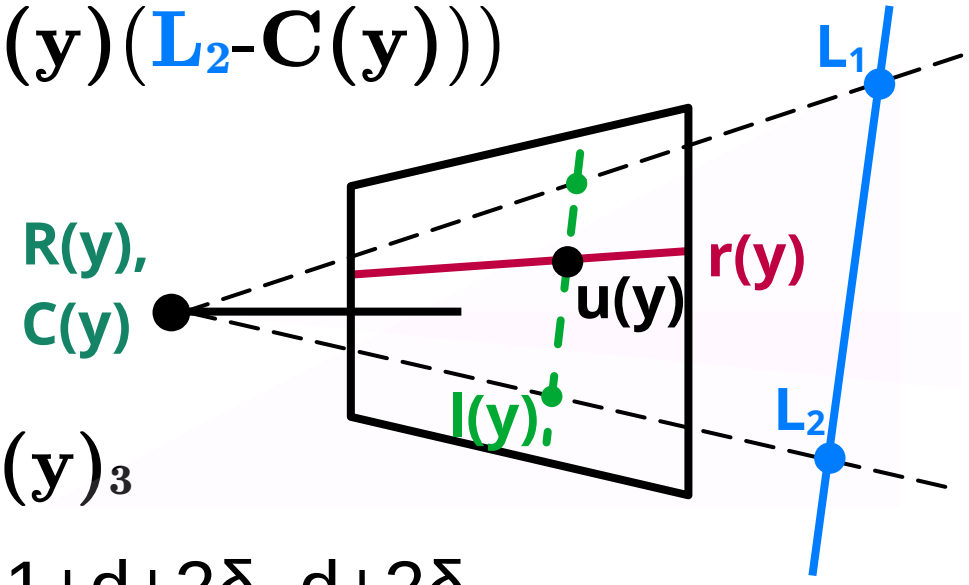
- $\mathbf{l(y)} = (\mathbf{R(y)}(\mathbf{L_1}\text{-}\mathbf{C(y)})) \times (\mathbf{R(y)}(\mathbf{L_2}\text{-}\mathbf{C(y)}))$



- $\mathbf{u(y)} = \mathbf{r(y)} \times \mathbf{l(y)}$
  - x coordinate of $\mathbf{u(y)}$: $\mathbf{u(y)_1}/\mathbf{u(y)_3}$
  - *rational function* of y, degrees 1+d+2δ, d+2δ
  - line projections are *algebraic curves* w. **2(1+d+2δ) DoF**

Projecting Lines

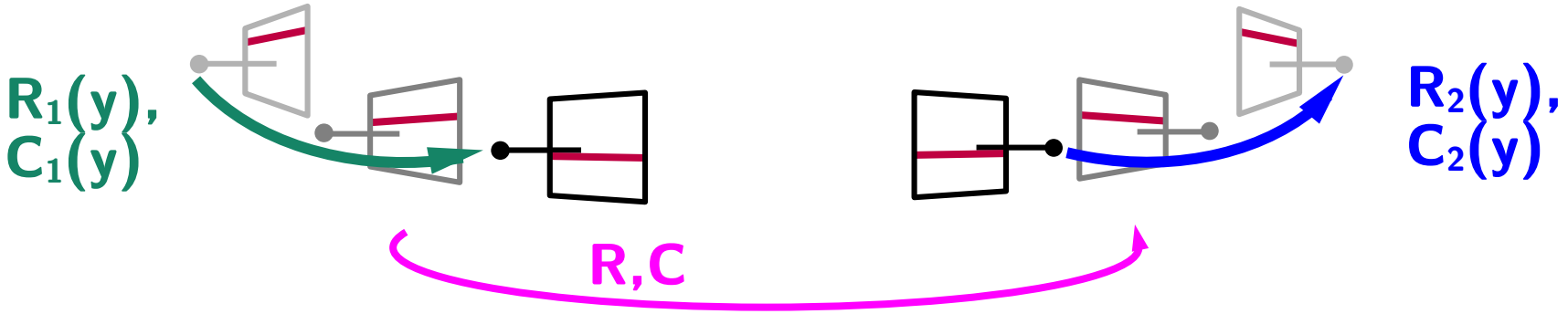
- $\mathbf{l(y)} = (\mathbf{R(y)}(\mathbf{L_1}\text{-}\mathbf{C(y)})) \times (\mathbf{R(y)}(\mathbf{L_2}\text{-}\mathbf{C(y)}))$



- $\mathbf{u(y)} = \mathbf{r(y)} \times \mathbf{l(y)}$
  - x coordinate of $\mathbf{u(y)}$: $\mathbf{u(y)_1}/\mathbf{u(y)_3}$
  - *rational function* of y, degrees 1+d+2δ, d+2δ
  - line projections are *algebraic curves* w. **2(1+d+2δ) DoF**
- Constraint $\mathbf{u(y)}^{\mathbf{T}}\mathbf{R(y)}[\mathbf{L_2}\text{-}\mathbf{L_1}]_{\times}(\mathbf{L_1}\text{-}\mathbf{C(y)})=\mathbf{0}$

RS Relative Pose Estimation

- **Input**: 2 RS images
- **Output**: camera pose + velocities + angular velocities



Minimal Problems

- Pose estimation ~ polynomial system $f(\theta,x)=0$
  - Minimal problem: finite, nonzero number of solutions [1]

[1] T. Duff, K. Kohn, A. Leykin, T. Pajdla: PLMP – Point-Line Minimal Problems in Complete Multi-View Visibility

Minimal Problems

- Pose estimation ~ polynomial system $f(\theta,x)=0$
  - Minimal problem: finite, nonzero number of solutions [1]
- Example (relative pose for calibrated GS cameras)
  - 5 points: 10 solutions
  - 4 points: ∞ solutions
  - 6 points: 0 solutions

[1] T. Duff, K. Kohn, A. Leykin, T. Pajdla: PLMP – Point-Line Minimal Problems in Complete Multi-View Visibility

Minimal Problems

- Pose estimation ~ polynomial system $f(\theta,x)=0$
  - Minimal problem: finite, nonzero number of solutions [1]
- Example (relative pose for calibrated GS cameras)
  - 5 points: 10 solutions
  - 4 points: ∞ solutions
  - 6 points: 0 solutions
- Necessary condition (Balanced problem [1]):
  - **Number of unknowns = Number of independent constraints**

[1] T. Duff, K. Kohn, A. Leykin, T. Pajdla: PLMP – Point-Line Minimal Problems in Complete Multi-View Visibility

Minimal Problems

- Pose estimation ~ polynomial system $f(\theta,x)=0$
  - Minimal problem: finite, nonzero number of solutions [1]
- Example (relative pose for calibrated GS cameras)
  - 5 points: 10 solutions
  - 4 points: ∞ solutions
  - 6 points: 0 solutions
- Necessary condition (Balanced problem [1]):
  - **Number of unknowns = Number of independent constraints**
- **fewer solutions -> faster solvers**

[1] T. Duff, K. Kohn, A. Leykin, T. Pajdla: PLMP – Point-Line Minimal Problems in Complete Multi-View Visibility

RS Relative Pose Estimation

- **Input**: 2 RS images
- **Output**: camera pose + velocities + angular velocities

RS Relative Pose Estimation

- **Input**: 2 RS images
- **Output**: camera pose + velocities + angular velocities
- Assume *constant velocity* + *constant angular velocity*
  - 5+6+6=**17 DoF**; we need **17 point correspondences**

RS Relative Pose Estimation

- **Input**: 2 RS images
- **Output**: camera pose + velocities + angular velocities
- Assume *constant velocity* + *constant angular velocity*
  - 5+6+6=**17 DoF**; we need **17 point correspondences**
- General approach: using point correspondences
  - We need 17 points.
  - Even with linearized rotation model > 1M solutions
  - Unsolvable in real time

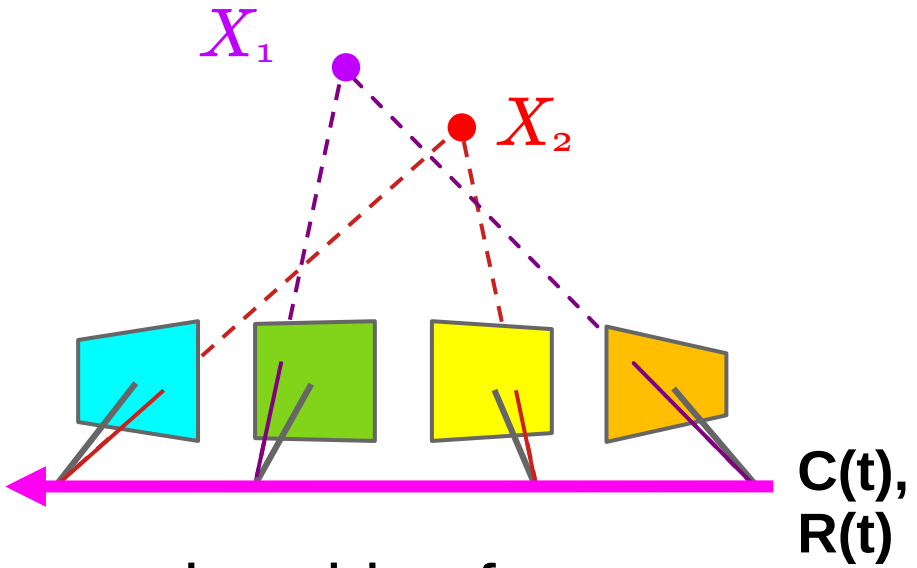
RS Relative Pose Estimation

- **Input**: 2 RS images
- **Output**: camera pose + velocities + angular velocities
- General approach UNSOLVABLE
- We need to use tricks:
  - **Assume common trajectory** [1]
  - **Use single scanline per image** [2]
    (no need to model trajectory)
  - **Estimate the pose from a single image** [3]
  - **Do the tricks from** [4]

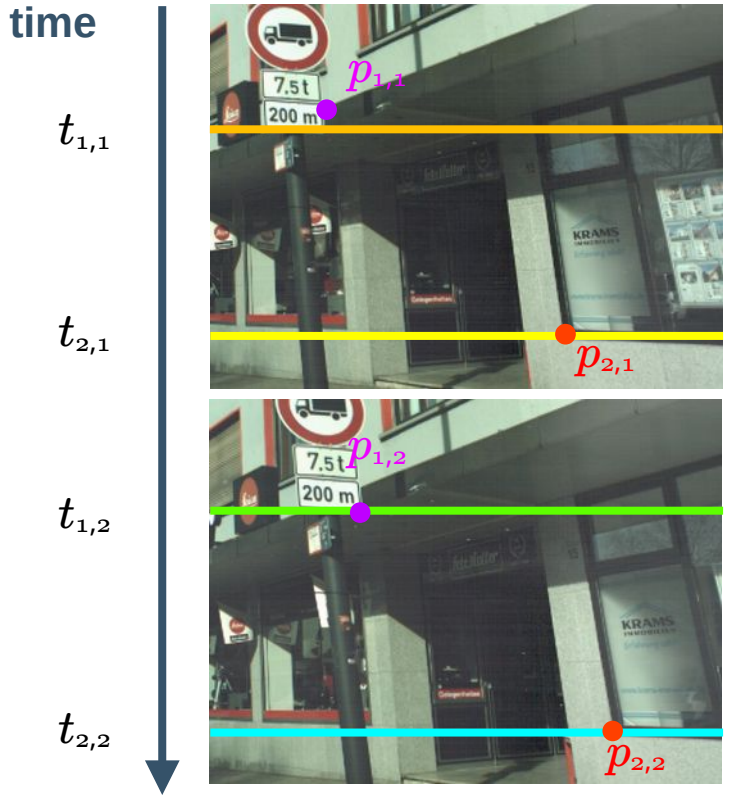
[1] P.Hruby, M.Pollefeys: Minimal Solvers for Full DoF Motion Estimation from Asynchronous Tracks, 3DV 2026
[2] P.Hruby, M.Pollefeys: Single-scanline relative pose estimation for rolling shutter cameras, ICCV 2025
[3] S. Errazuriz Munoz, K.Kiehn, P.Hruby. K.Kohn: Single-View Rolling-Shutter SfM
[4] D.Barath: Rolling Shutter Relative Pose Estimation Made Practical

Assume Common Trajectory

Minimal Solvers for Full DoF Motion Estimation from Asynchronous Tracks

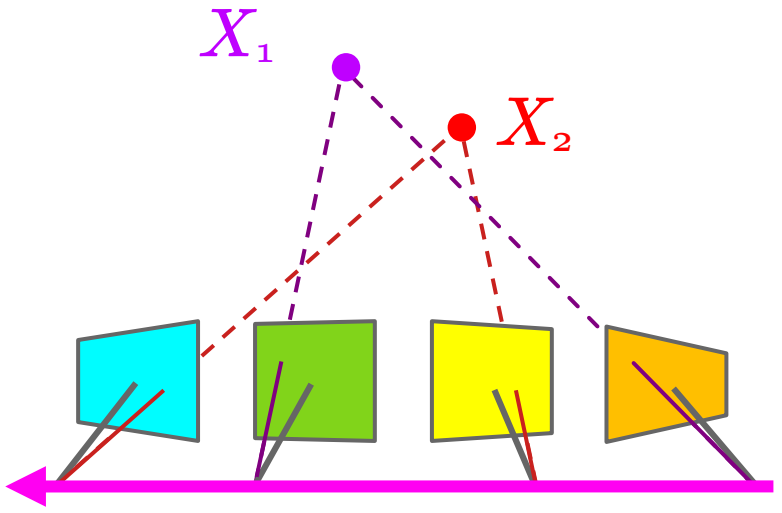


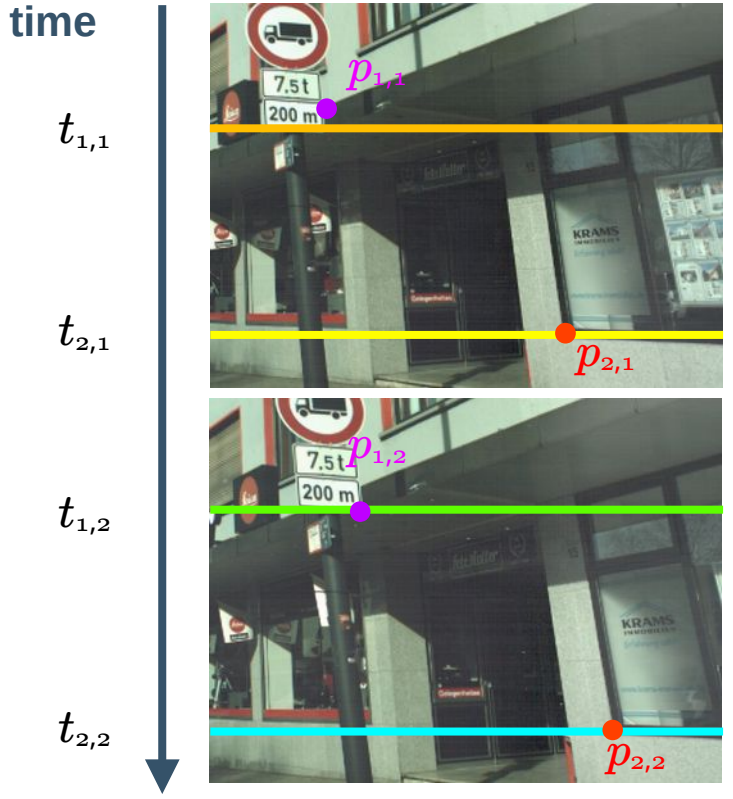
- consecutive video frames
- constant velocity & angular velocity



Assume Common Trajectory

Minimal Solvers for Full DoF Motion Estimation from Asynchronous Tracks

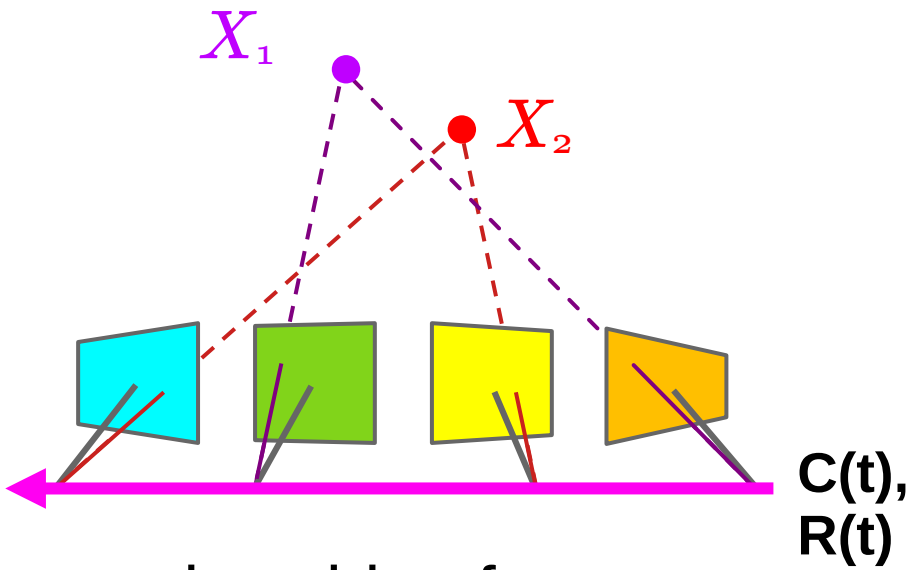


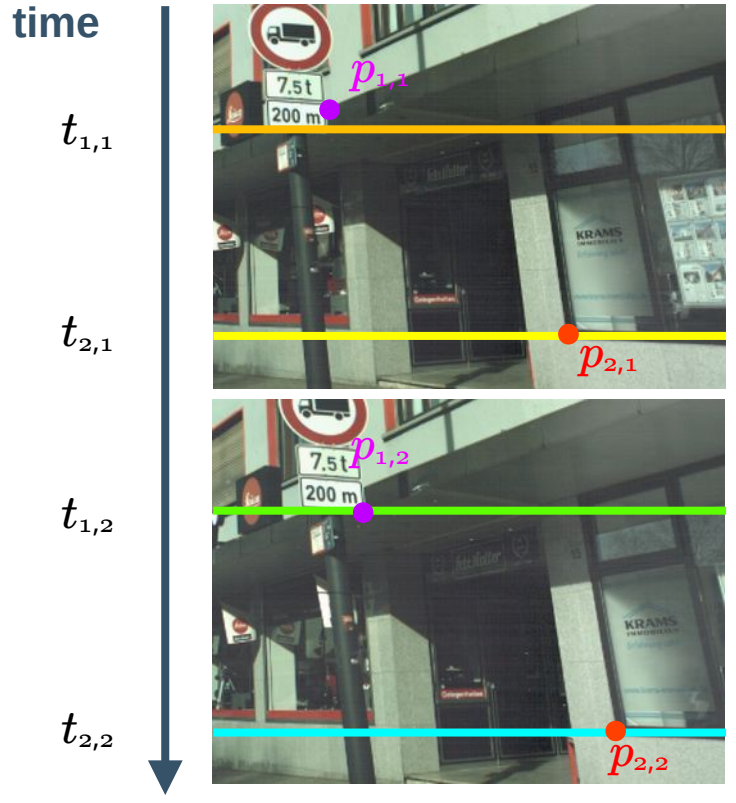


- consecutive video frames
- constant velocity & angular velocity
- sharing parameters: 5 variables

Assume Common Trajectory

Minimal Solvers for Full DoF Motion Estimation from Asynchronous Tracks





- consecutive video frames
- constant velocity & angular velocity
- sharing parameters: 5 variables
- 20 solutions with linearized rotation

Assume Common Trajectory

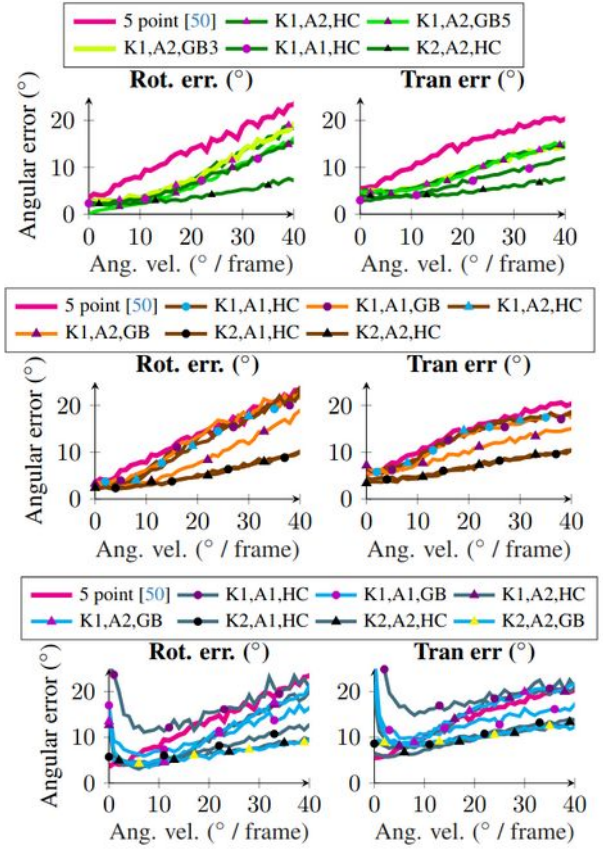
Minimal Solvers for Full DoF Motion Estimation from Asynchronous Tracks

SOLVERS

K	Approximation 1 m=2	m=3	m=4	Approximation 2 m=2	m=3	m=4
1	120	22	2	20	20	8
2	426	120	36	122	102	36
3	732	210	64	276	196	64
4	1038	300	92	430	290	92
5	1344	390	120	584	384	120
6	1650	480	148	738	478	148

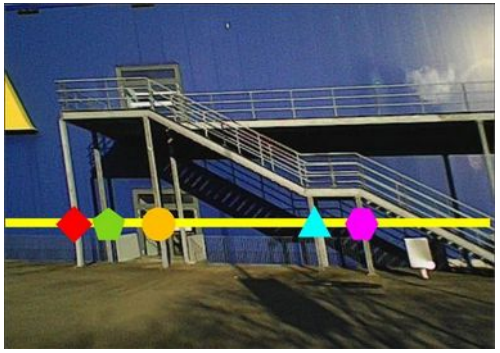
m,n	K	appx	type	#vars	#sols	time (μs)
2,5	1	A1	HC	5	120	1.8E5
2,5	1	A2	HC	5	20	5.2E3
2,5	1	A2	GB	5	20	4.7E2
2,5	1	A2	GB	3	20	**1.2E2**
2,5	2	A2	HC	5	122	2.1E5
3,2	1	A1	HC	11	22	1.6E4
3,2	1	A1	GB	5	22	**2.1E3**
3,2	1	A2	HC	11	20	1.2E4
3,2	1	A2	GB	5	20	2.3E3
3,2	2	A1	HC	11	120	3.4E5
3,2	2	A2	HC	11	120	2.1E5
4,1	1	A1	HC	5	2	6.7E2
4,1	1	A1	GB	5	8	**8.1E1**
4,1	1	A2	HC	8	8	3.2E3
4,1	1	A2	GB	8	8	1.0E2
4,2	2	A1	HC	8	36	6.6E4
4,2	2	A2	HC	8	36	2.8E4
4,2	2	A2	GB	5	36	3.1E4

EXPERIMENTS



Use single scanline per image

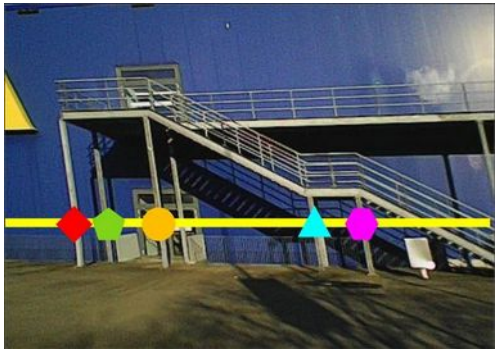
Single-scanline relative pose estimation for rolling shutter cameras



- Pixels at this scanline captured at the same time
  - No need to model the camera trajectory
- **Features**: intersections of line projections with the scanline
- **Output:** Poses of the scanlines, 3D lines

Use single scanline per image

Single-scanline relative pose estimation for rolling shutter cameras



- Pixels at this scanline captured at the same time
  - No need to model the camera trajectory
- **Features**: intersections of line projections with the scanline
- **Output:** Poses of the scanlines, 3D lines
- Constraint: $\mathbf{u}(\mathbf{y})^{\mathrm{T}}\mathbf{R}(\mathbf{y})[\mathbf{L}_2\text{-}\mathbf{L}_1]_{\times}(\mathbf{L}_1\text{-}\mathbf{C}(\mathbf{y}))=\mathbf{0}$

Use single scanline per image

Single-scanline relative pose estimation for rolling shutter cameras

#cameras m, #lines n

- A. Fully generic setting
  - Minimal problems: m=5,n=23; m=21,n=7

Use single scanline per image

Single-scanline relative pose estimation for rolling shutter cameras

#cameras m, #lines n

- A. Fully generic setting
  - Minimal problems: m=5,n=23; m=21,n=7
  - We could not determine the number of solutions.

Use single scanline per image

Single-scanline relative pose estimation for rolling shutter cameras

#cameras m, #lines n

- A. Fully generic setting
  - Minimal problems: m=5,n=23; m=21,n=7
  - We could not determine the number of solutions.
- B. Parallel lines
  - Up to a projective transformation of the scene
- C. Gravity prior for cameras + Generic lines
- D. Gravity prior for cameras + Parallel lines
- E. Gravity prior for cameras + Vertical lines

S.	m	n	Min.	Deg
A	5	23	Y	389k+
A	21	7	Y	40k+
B	3	7	Y	2^*
B	4	6	Y	2^*
C	5	15	Y	1.8M+
C	15	5	Y	532k+

S.	m	n	Min.	Deg
D	3	7	Y	48
D	4	5	Y	232
D	6	4	Y	1224
E	3	5	Y	16
E	4	4	Y	32

Use single scanline per image

Single-scanline relative pose estimation for rolling shutter cameras

#cameras m, #lines n

- A. Fully generic setting
  - Minimal problems: m=5,n=23; m=21,n=7
  - We could not determine the number of solutions.
- B. Parallel lines
  - Up to a projective transformation of the scene
  - Solved by transforming to 1D pose estimation
- C. Gravity prior for cameras + Generic lines
- D. Gravity prior for cameras + Parallel lines
- E. Gravity prior for cameras + Vertical lines
  - Solved by transforming to 1D pose estimation

S.	m	n	Min.	Deg
A	5	23	Y	389k+
A	21	7	Y	40k+
B	3	7	Y	2*
B	4	6	Y	2*
C	5	15	Y	1.8M+
C	15	5	Y	532k+

S.	m	n	Min.	Deg
D	3	7	Y	48
D	4	5	Y	232
D	6	4	Y	1224
E	3	5	Y	16
E	4	4	Y	32

Use single scanline per image

Single-scanline relative pose estimation for rolling shutter cameras

Results: Fastec dataset

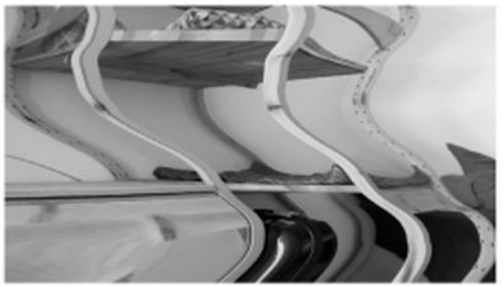
	solver	$\leq 10°$	$\leq 20°$	$\leq 30°$
MV	(E,3,5)	1.0 %	6.2 %	21.9 %
	(E,4,4)	1.9 %	7.9 %	**22.4** %
	(D,3,7)	**5.6** %	**13.8** %	21.7 %
SV	(E,3,5)	0.5 %	2.5 %	5.3 %
	(E,4,4)	1.2 %	4.1 %	10.0 %
	(D,3,7)	0.5 %	2.2 %	6.6 %

Table 3. Percentage of camera poses from Fastec [43] with pose error below $10/20/30°$. MV: *Multi-view*, SV: *Single-view*.

	solver	$\leq 5°$	$\leq 10°$	$\leq 20°$	$\leq 30°$
MV	(E,3,5)	1	3	10	15
	(E,4,4)	5	8	11	14
	(D,3,7)	**6**	**10**	**15**	**17**
SV	(E,3,5)	3	3	5	10
	(E,4,4)	2	4	7	13
	(D,3,7)	4	4	7	9

Table 2. Number of sequences (out of 19) from the Fastec dataset [43], with at least one estimated pose with error below $5/10/20/30°$. MV: *Multi-view*, SV: *Single-view*.

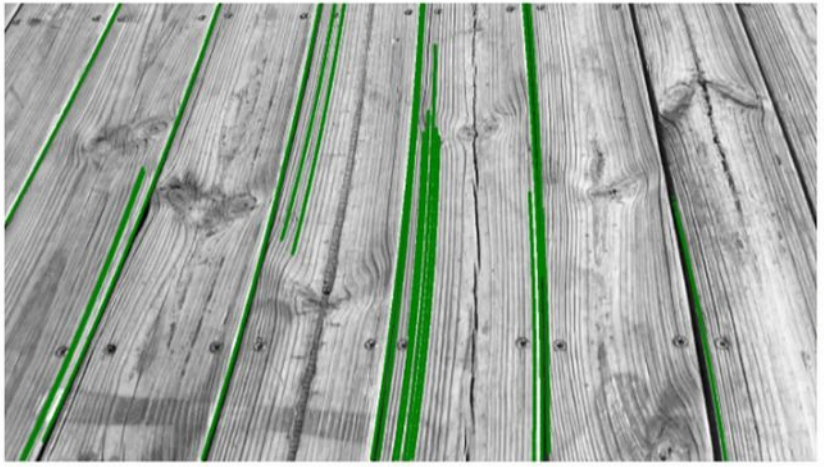
Results: Custom dataset



- median rotation and translation error: 4.8˚ and 14.8˚

Estimate the pose from a single image

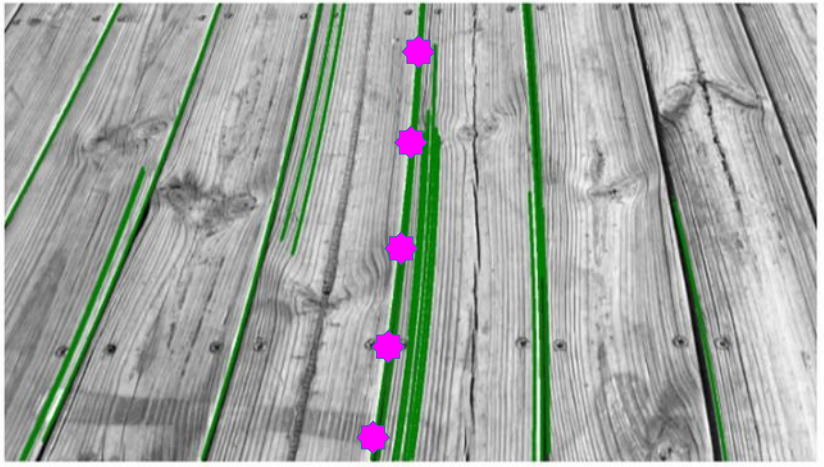
Single-View Rolling Shutter SfM



- **Input**: detected curves - projections of lines
- **Variables**: motion parameters, 3D lines

Estimate the pose from a single image

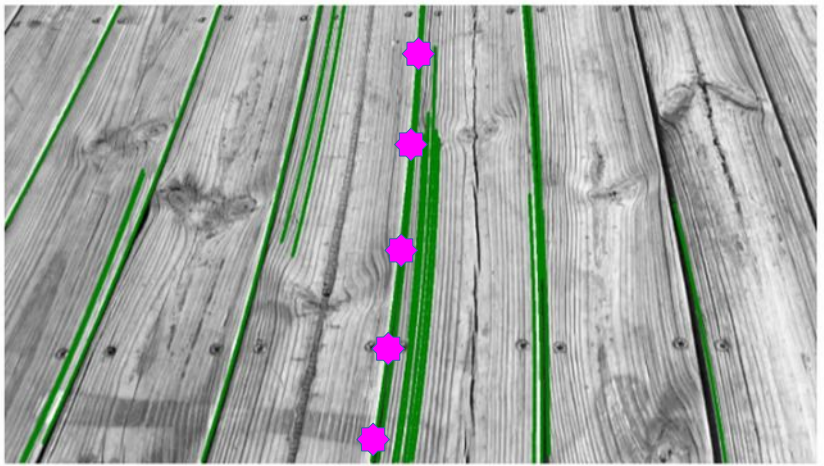
Single-View Rolling Shutter SfM



- **Input**: detected curves - projections of lines
- **Variables**: motion parameters, 3D lines
- **Measurements**: points on the line projections

Estimate the pose from a single image

Single-View Rolling Shutter SfM



- **Input**: detected curves - projections of lines
- **Variables**: motion parameters, 3D lines
- **Measurements**: points on the line projections
- **Constraints**: $\mathbf{u(y)^T R(y)[L_2\text{-}L_1]_\times (L_1\text{-}C(y))=0}$

Estimate the pose from a single image

Single-View Rolling Shutter SfM

- **Number of unknowns = Number of independent constraints**

Estimate the pose from a single image

Single-View Rolling Shutter SfM

- **Number of unknowns = Number of independent constraints**
- **Motion unknowns**
- **Structure unknowns**
- **Constraints**

Estimate the pose from a single image

Single-View Rolling Shutter SfM

- **Number of unknowns = Number of independent constraints**
- **Motion unknowns** C(y): degree d, R(y): Cayley, degree δ
  - Generic lines: 3(d+δ)-1 DoF
- **Structure unknowns**

- **Constraints**

Estimate the pose from a single image

Single-View Rolling Shutter SfM

- **Number of unknowns = Number of independent constraints**
- **Motion unknowns** C(y): degree d, R(y): Cayley, degree δ
  - Generic lines: 3(d+δ)-1 DoF
- **Structure unknowns** (ℓ lines):
  - Generic lines: 4ℓ DoF

- **Constraints**

Estimate the pose from a single image

Single-View Rolling Shutter SfM

- **Number of unknowns = Number of independent constraints**
- **Motion unknowns** C(y): degree d, R(y): Cayley, degree δ
  - Generic lines: 3(d+δ)-1 DoF
- **Structure unknowns** (ℓ lines):
  - Generic lines: 4ℓ DoF

- **Constraints** (N_i points per line i)
  - Fix ΣN_i DoF
  - At most 2(1+d+2δ) independent points per line

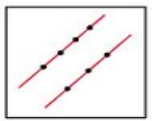
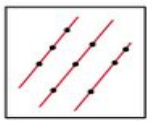
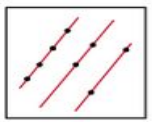
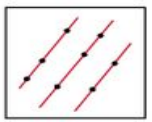
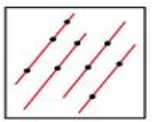
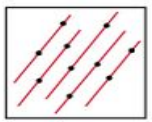
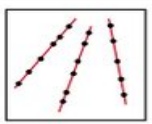
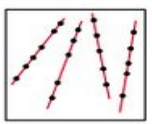
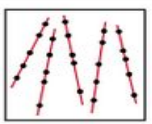
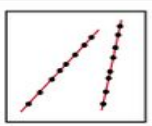
Estimate the pose from a single image

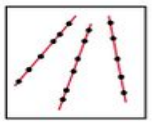
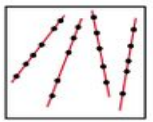
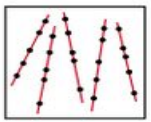
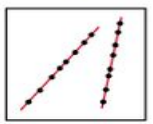
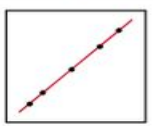
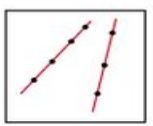
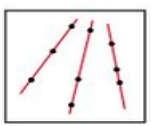
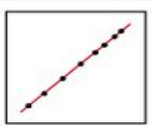
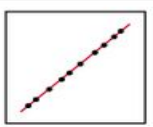
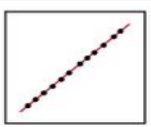
Single-View Rolling Shutter SfM

- **Number of unknowns = Number of independent constraints**
- **Motion unknowns** C(y): degree d, R(y): Cayley, degree δ
  - Generic lines: 3(d+δ)-1 DoF, Parallel lines: 2d+3δ-1 DoF
- **Structure unknowns** (ℓ lines):
  - Generic lines: 4ℓ DoF
  - Parallel lines: 2ℓ+2 DoF
  - Parallel + coplanar lines: ℓ+3 DoF
- **Constraints** (N_i points per line i)
  - Fix ΣN_i DoF
  - At most 2(1+d+2δ) independent points per line

Estimate the pose from a single image

Single-View Rolling Shutter SfM

$\delta = 0,\ d = 1$				
lable	$d_1(4,3)P$	$d_1(3^3)P$		
degree	2	5		
$\delta = 0,\ d = 1$				
lable	$d_1(4,2^2)PC$	$d_1(3^2,2)PC$	$d_1(3,2^3)PC$	$d_1(2^5)PC$
degree	2	4	6	10
$\delta = 0,\ d = 2$				
lable	$d_2(6^2,5)$	$d_2(6,5^3)$	$d_2(5^5)$	
degree	30	131	680	
$\delta = 0,\ d = 3$		14 further balanced problems; see Example 8		

$\delta = 0,\ d = 2$					
lable	$d_2(6^2,5)$	$d_2(6,5^3)$	$d_2(5^5)$		
degree	30	131	680		
$\delta = 0,\ d = 3$		14 further balanced problems; see Example 8			
lable	$d_3(8^2)$				
degree	25				
$\delta = 1,\ d = 0$					
lable	$\delta_1(5)$	$\delta_1(4,3)$	$\delta_1(3^3)$		
degree	10	30	54		
$\delta = 1,\ 0 < d \leq 5$					
lable	$d_1\delta_1(8)$	$d_2\delta_1(10)$	$d_3\delta_1(12)$	$d_4\delta_1(14)$	$d_5\delta_1(16)$
degree	20	104	320	760	1540

Estimate the pose from a single image

Single-View Rolling Shutter SfM

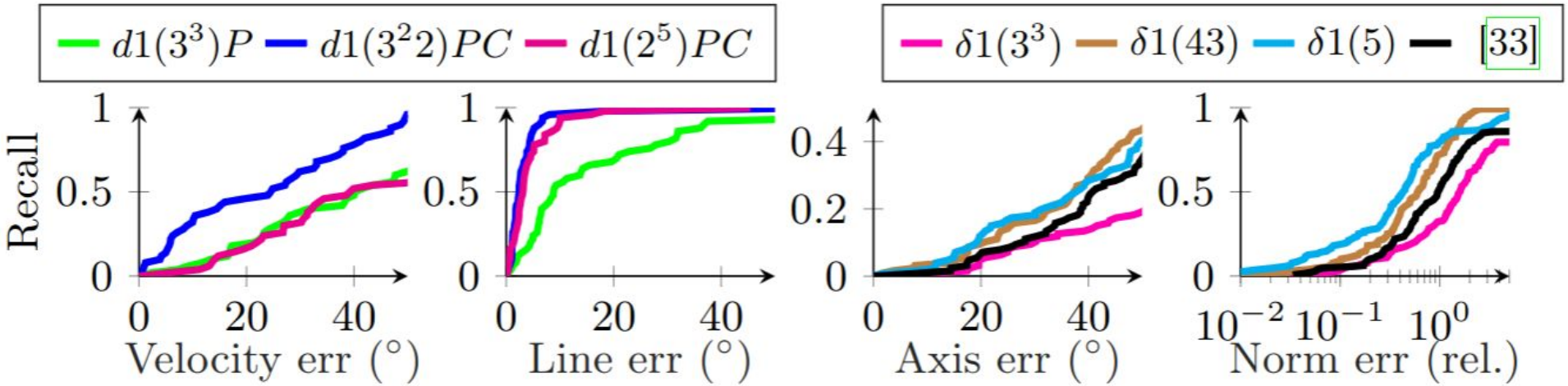
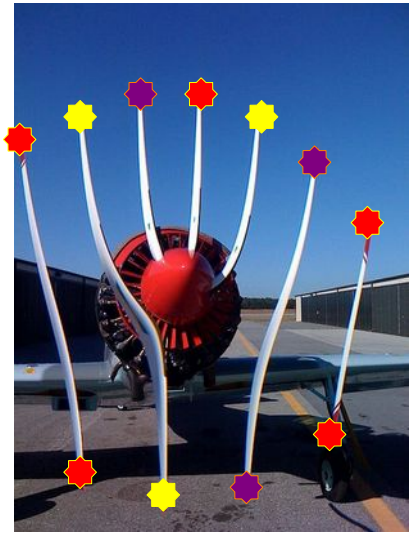


Fig. 4: *Real-world experiments.* Recall curves of the velocity and line direction errors of the proposed solvers with $\delta = 0$ on the dataset [26] (*left*), and of the axis and norm errors of the proposed solvers with $d = 0$ on the dataset [19].

Point SfM



- **Input**: multiple observations of the same point
- **Unknowns**: motion parameters, 3D points

Point SfM: Constraints

- A point $\mathbf{X}$ is observed $k \leq 1+d+2\delta$ times by the camera
  - This imposes 2k constraints on the structure and motion
  - Constraint: $\mathbf{u}_i \sim \mathbf{R}(\mathbf{y}_i) \cdot (\mathbf{X} - \mathbf{C}(\mathbf{y}_i))$

Point SfM: Constraints

- A point $\mathbf{X}$ is observed $k \leq 1+d+2\delta$ times by the camera
  - This imposes 2k constraints on the structure and motion
  - Constraint: $\mathbf{u}_i \sim \mathbf{R}(\mathbf{y}_i) \cdot (\mathbf{X} - \mathbf{C}(\mathbf{y}_i))$
- How many points do we need to find the structure and motion?
  - When observing the point **$1+d+2\delta$** times?
  - When observing the point **2** times?

Point SfM: Minimal Problems

- Considering maximal number of projections:

Theorem

Consider a generic RS camera in $\mathcal{P}_{d,\delta}$ observing p world points. The balanced SfM problems using all $1 + d + 2\delta$ image points per world point are:

- $p = 2, d = 2, \delta = 0$: *this is minimal of degree 1;*
- $p = 2, d = 1, \delta = 0$: *this is minimal of degree 1;*
- $p = 1, d = \delta$: *minimal for $d \leq 5$, with degrees 16 for $d = 1$ and 904 for $d = 2$.*

Point SfM: Minimal Problems

- Considering 2 projections:

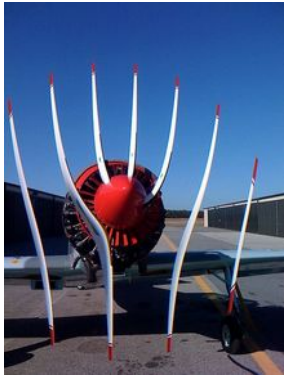
Theorem

Consider a generic RS camera in $\mathcal{P}_{d,\delta}$ observing p world points. The balanced SfM problems using 2 image points per world point are:

- $d > 0, 3(d+\delta) - 1 = p$: *this is minimal of degree 1 whenever* $\delta = 0$, *minimal of degree* 140 *for* $d = \delta = 1$, *and minimal of degree* 804 *for* $d = 2, \delta = 1$;
- $d = 0, 3\delta = 2p$: *this is minimal of degree* 112 *for* $\delta = 2, p = 3$.

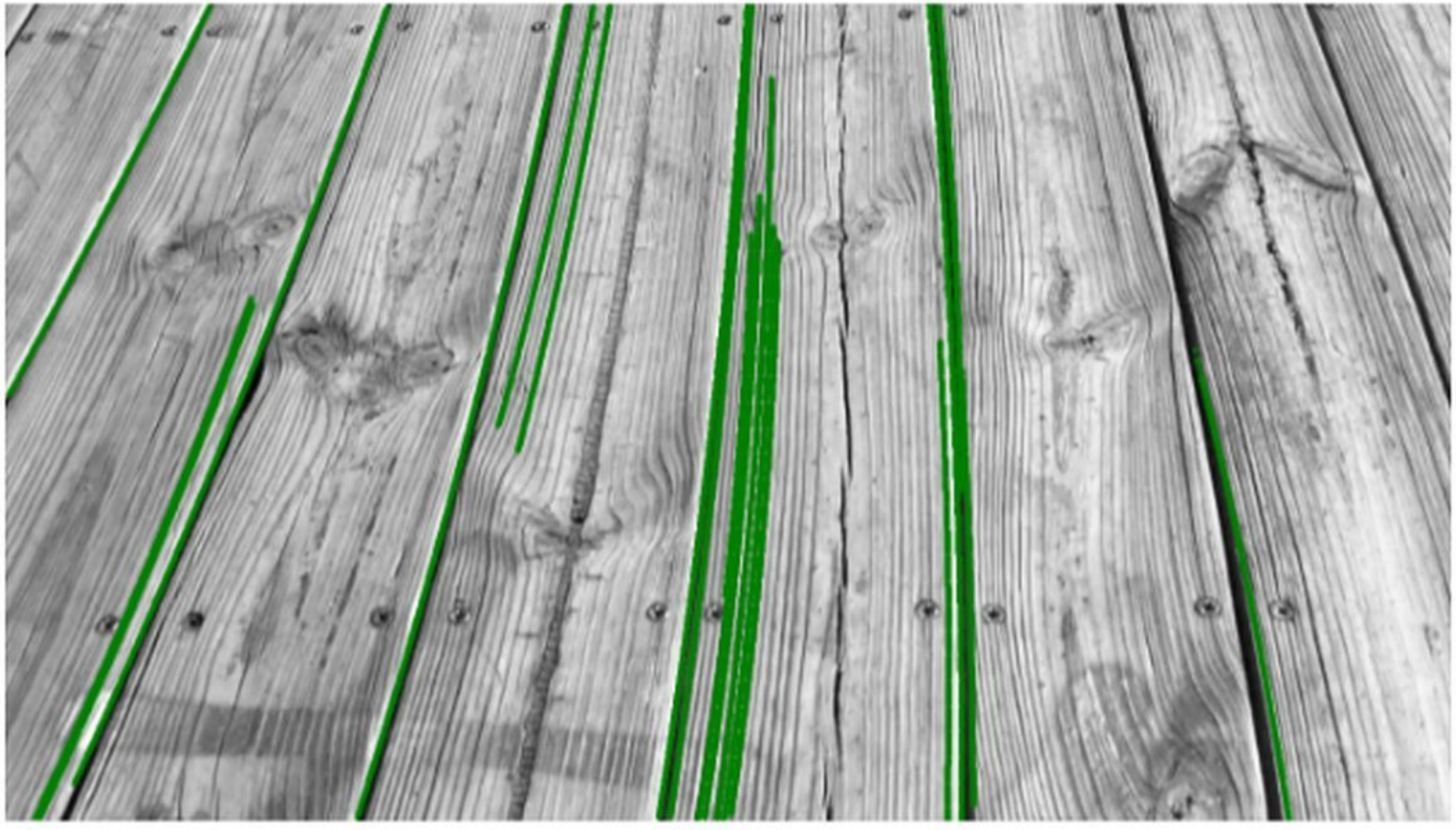
Detecting and Matching Features

- **Points:**
  - Detected and matched with standard methods.
  - Can the performance be improved by assuming RS distortion, e.g. in the training data?
  - Can we detect multiple observations in single image?

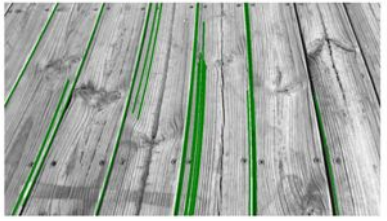


- **Lines (small distortion):**
  - detected with DeepLSD, matched with GlueStick

Detecting+Matching Distorted Lines

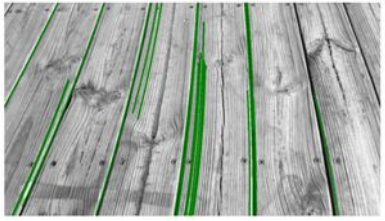


Detecting+Matching Distorted Lines



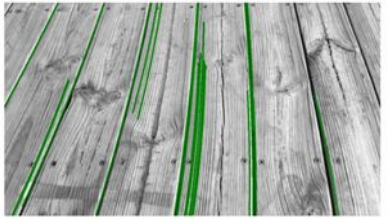
Detecting+Matching Distorted Lines

- **Detection:**
  - 1. Canny edge detection
    - every edge point has coordinates + normal direction
  - 2. Sequential RANSAC
    - Fit curves into the detected edge points
    - Select the curves with the largest number of inliers



Detecting+Matching Distorted Lines

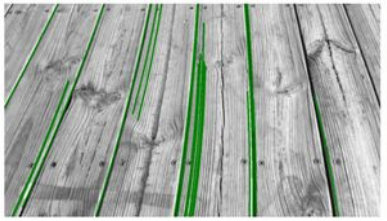
- **Detection:**
  - 1. Canny edge detection
    - every edge point has coordinates + normal direction
  - 2. Sequential RANSAC
    - Fit curves into the detected edge points
    - Select the curves with the largest number of inliers



- **Matching:**
  - 1. Use a dense matcher (i.e. RoMA) to match the images
  - 2. Every match between the curves increases the score for the pair
  - 3. Select the pairs with the best score

Detecting+Matching Distorted Lines

- **Detection:**
  - 1. Canny edge detection
    - every edge point has coordinates + normal direction
  - 2. Sequential RANSAC
    - Fit curves into the detected edge points
    - Select the curves with the largest number of inliers



- **Matching:**
  - 1. Use a dense matcher (i.e. RoMA) to match the images
  - 2. Every match between the curves increases the score for the pair
  - 3. Select the pairs with the best score
- Can this be improved with deep learning?

Future Challenges

- Study the general 17pt minimal problem
- Study further tricks to develop minimal solvers
- Use these solvers in 3D pipeline
  - Figure out, which solvers to use, which motion models work for which scenarios, ...
- Feature detection + matching
- Find actual practical usecases
  - where the benefits could make real change