Dense Continuous-Time Tracking and Mapping with Rolling Shutter RGB-D Cameras

Christian Kerl, Jörg Stückler and Daniel Cremers

Motivation

KinectFusion

DVO SLAM

Kintinuous

ElasticFusion

Motivation

KinectFusion

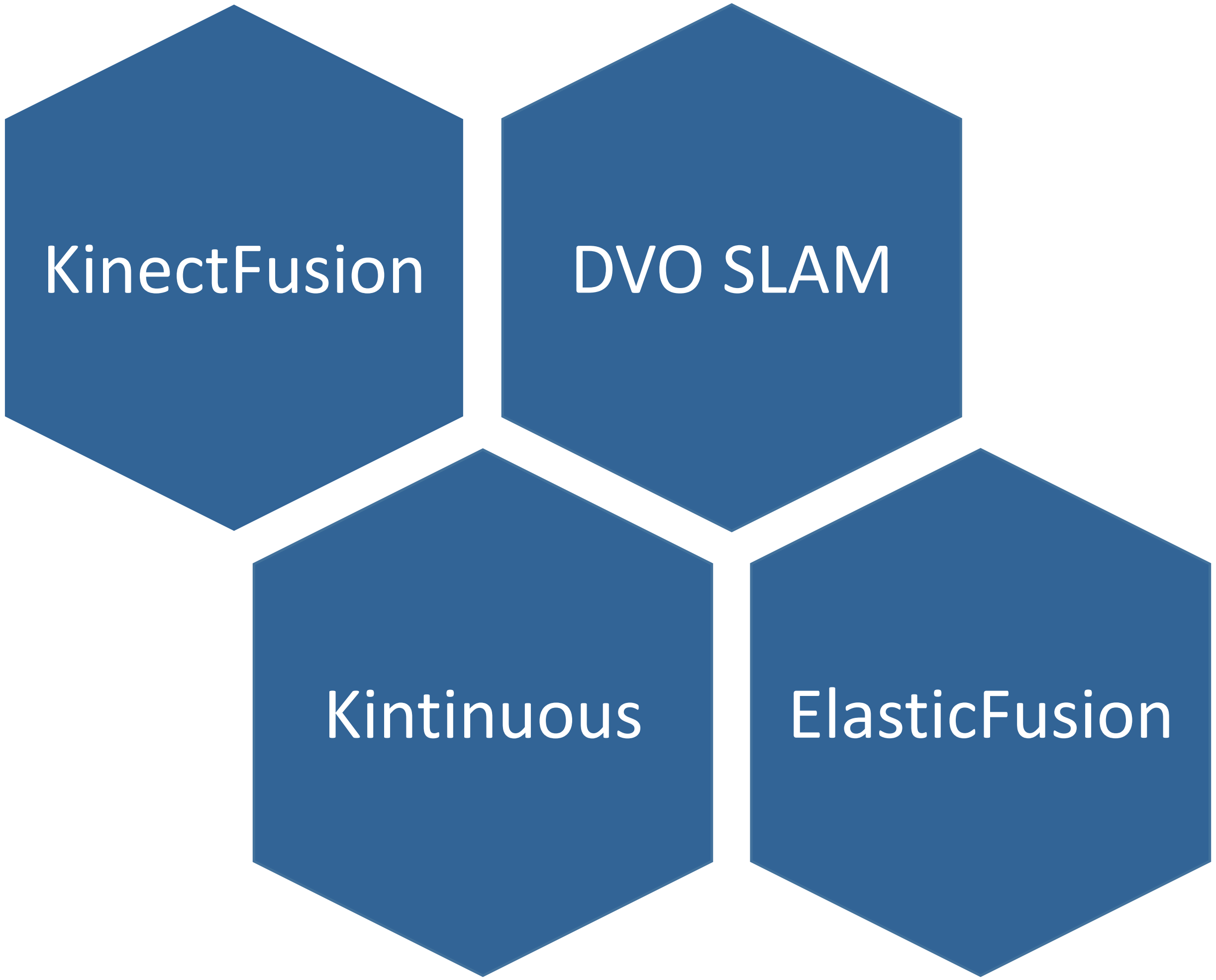
DVO SLAM

Kintinuous

ElasticFusion

What do they have in common?

Motivation



What do they have in common?

- discrete poses
- global shutter
- dense tracking

Motivation

KinectFusion

DVO SLAM

Kintinuous

ElasticFusion

What do they have in common?

- discrete poses
- global shutter
- dense tracking

This talk

- continuous trajectories
- rolling shutter
- dense tracking

Continuous Trajectory Representation

- Trajectory function

$$\boldsymbol{T}(t) : \mathbb{R} \rightarrow \mathrm{SE}(3)\,\forall\, t \in [t_{\mathsf{min}}, t_{\mathsf{max}})$$

Continuous Trajectory Representation

- Trajectory function

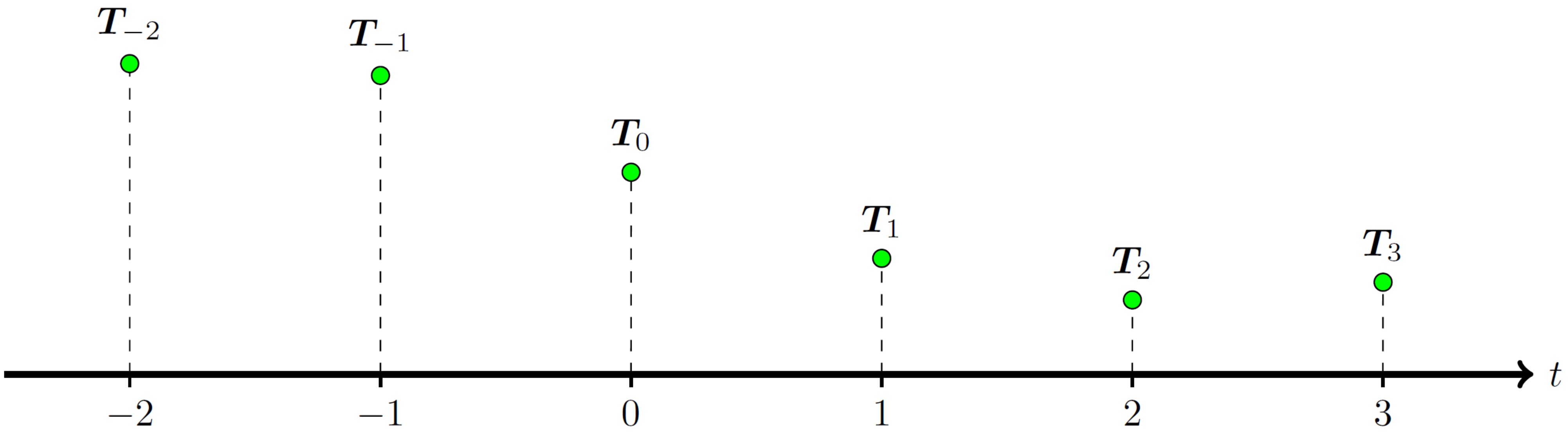
$$\boldsymbol{T}(t) : \mathbb{R} \rightarrow \mathrm{SE}(3)\, \forall\, t \in [t_{\text{min}}, t_{\text{max}})$$

- Cumulative B-Splines for SE3 (Lovegrove et al. BMVC2013)
- Few other representations in literature

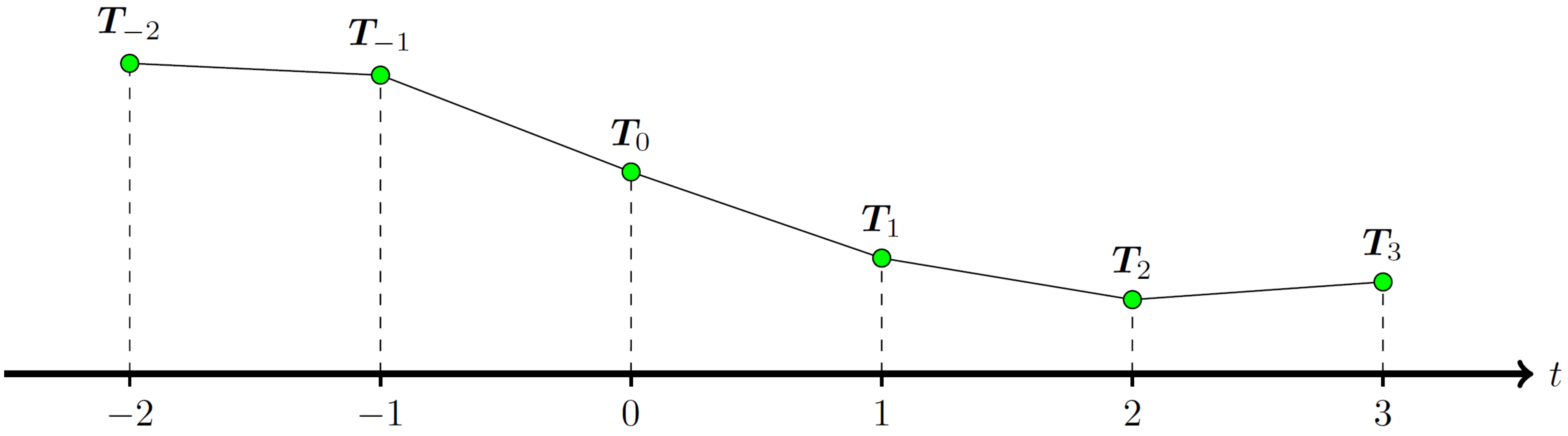
Advantages

- Combine multiple sensors
  - Different measurement rates
  - Unsynchronized
- Fewer variables than measurements
- Constrains motion (no jumping)
- Differentiable

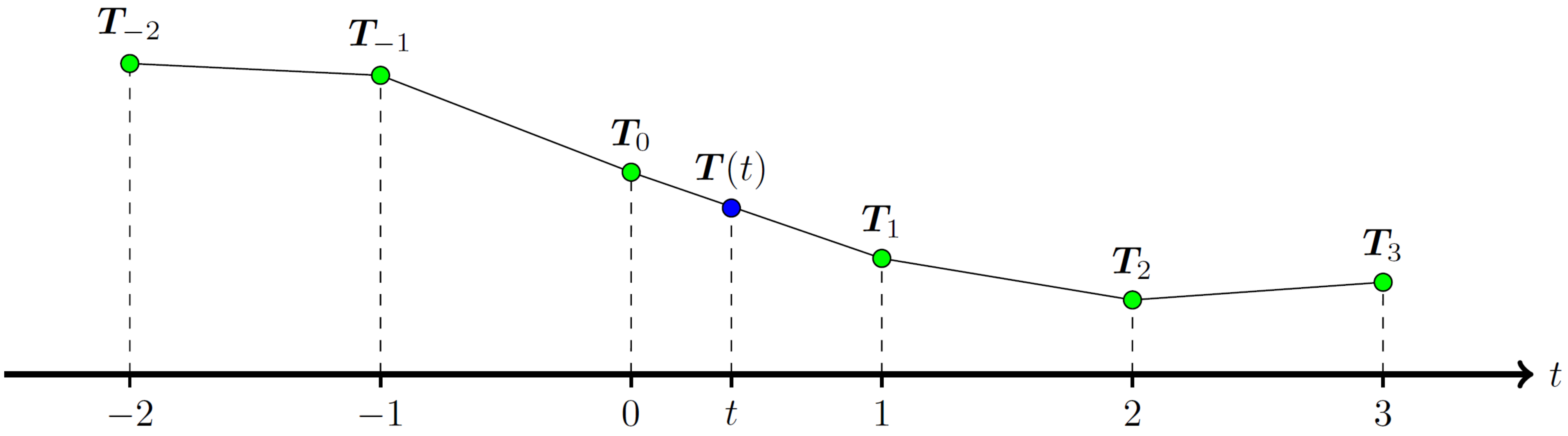
From Discrete to Continuous



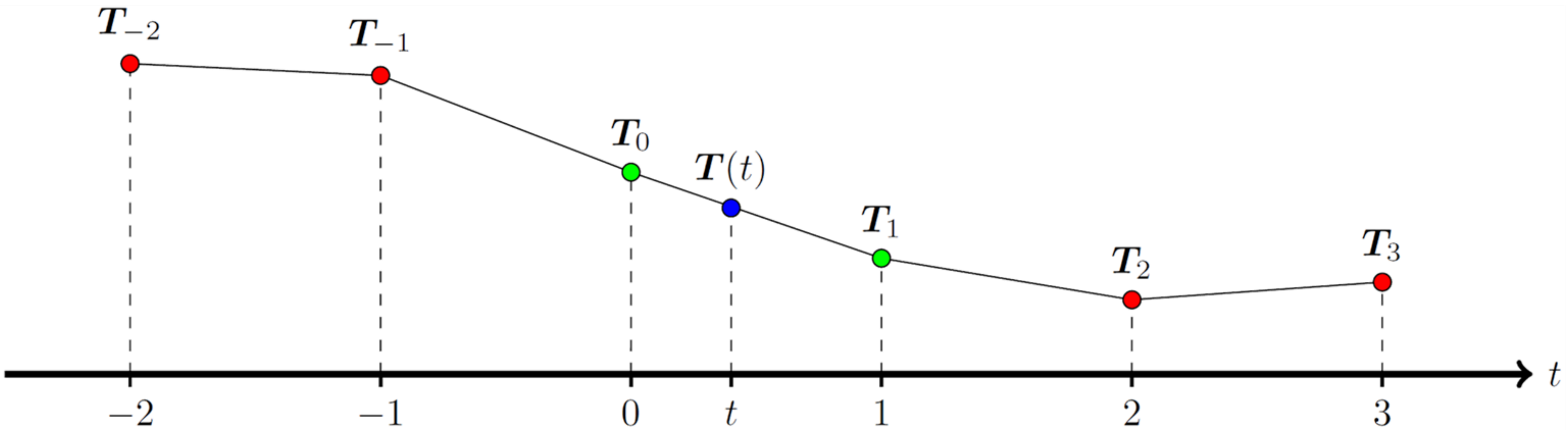
From Discrete to Continuous



From Discrete to Continuous

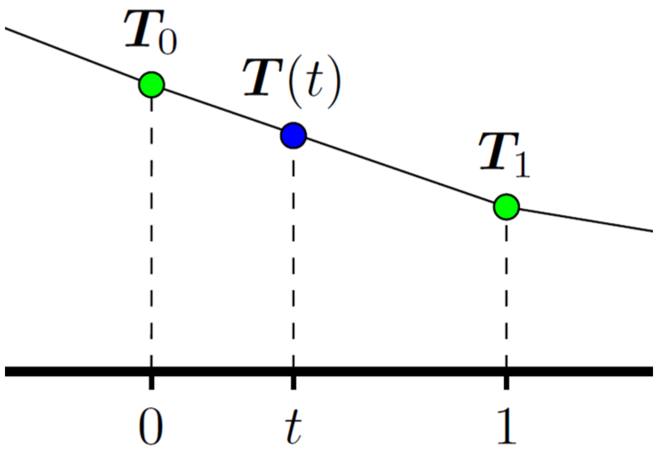


From Discrete to Continuous



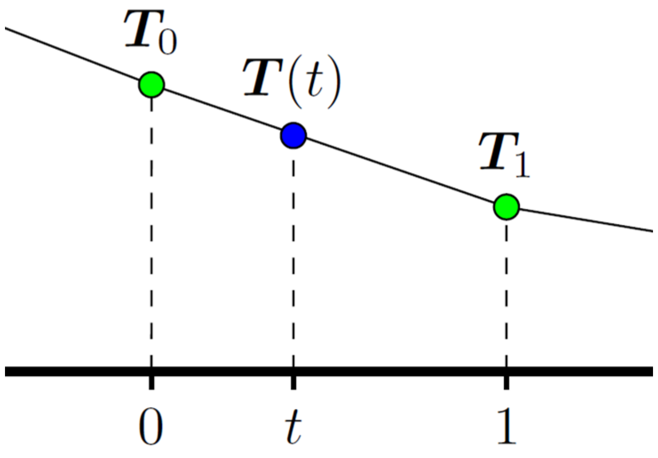
Cumulative Splines for SE3

$$\boldsymbol{T}(t) = (1-t)\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1$$



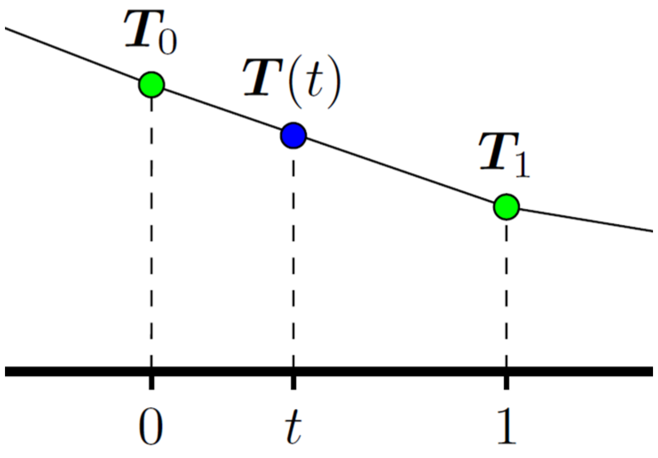
Cumulative Splines for SE3

$$
\begin{aligned}
\boldsymbol{T}(t) &= (1-t)\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1 \\
&= \boldsymbol{T}_0 - t\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1
\end{aligned}
$$



Cumulative Splines for SE3

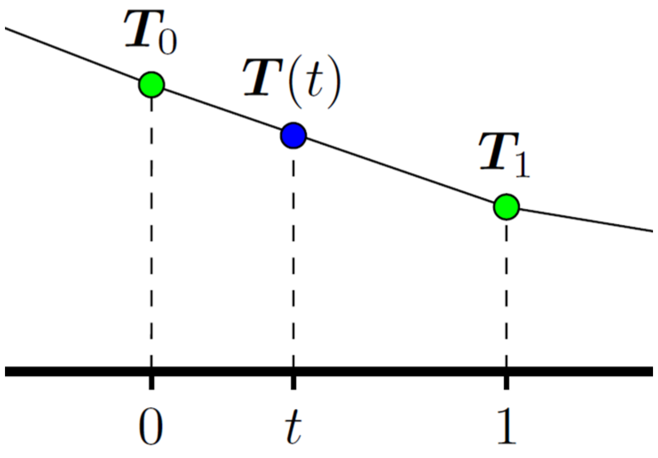
$$\begin{aligned} \boldsymbol{T}(t) &= (1-t)\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1 \\ &= \boldsymbol{T}_0 - t\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1 \\ &= \boldsymbol{T}_0 + t\,(\boldsymbol{T}_1 - \boldsymbol{T}_0) \end{aligned}$$



Cumulative Splines for SE3

$$
\begin{aligned}
\boldsymbol{T}(t) &= (1-t)\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1 \\
&= \boldsymbol{T}_0 - t\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1 \\
&= \boldsymbol{T}_0 + t\,(\boxed{\boldsymbol{T}_1 - \boldsymbol{T}_0})
\end{aligned}
$$

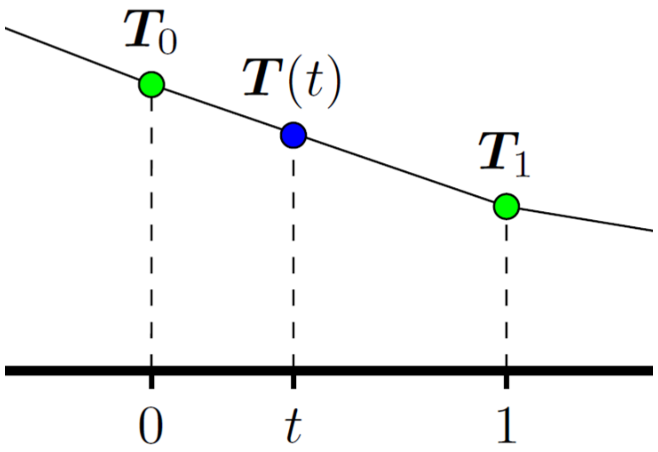
$$\boxed{\boldsymbol{T}_0^{-1}\boldsymbol{T}_1}$$



Cumulative Splines for SE3

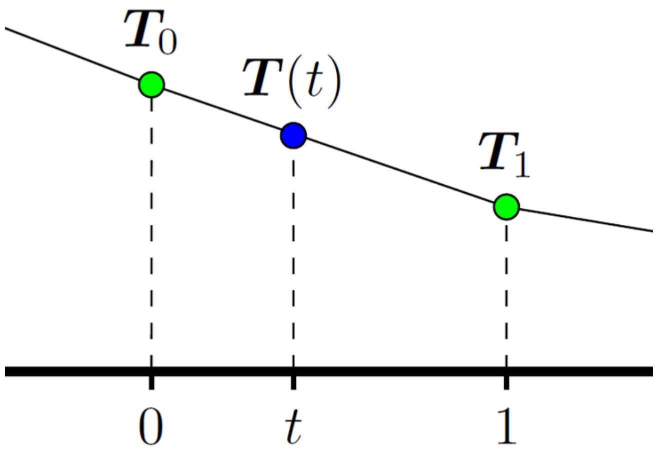
$$\begin{aligned}\boldsymbol{T}(t) &= (1-t)\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1\\ &= \boldsymbol{T}_0 - t\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1\\ &= \boldsymbol{T}_0 + t\,(\boldsymbol{T}_1 - \boldsymbol{T}_0)\end{aligned}$$

$$\log(\boldsymbol{T}_0^{-1}\boldsymbol{T}_1)$$



Cumulative Splines for SE3

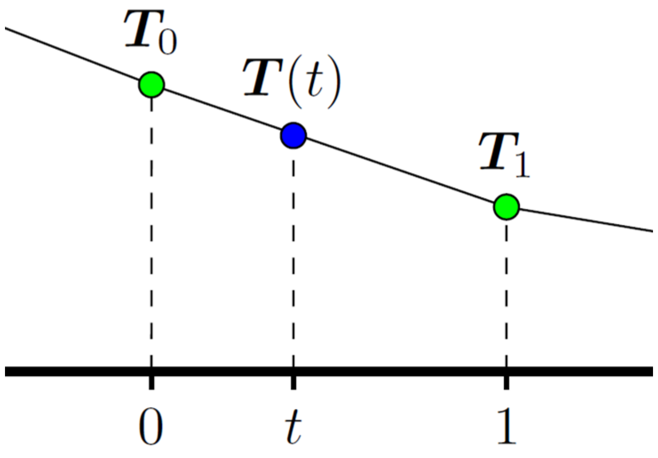
$$\boldsymbol{T}(t) = (1-t)\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1$$
$$= \boldsymbol{T}_0 - t\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1$$
$$= \boldsymbol{T}_0 + \boxed{t}(\boldsymbol{T}_1 - \boldsymbol{T}_0)$$
$$\boxed{t}\log(\boldsymbol{T}_0^{-1}\boldsymbol{T}_1)$$



Cumulative Splines for SE3

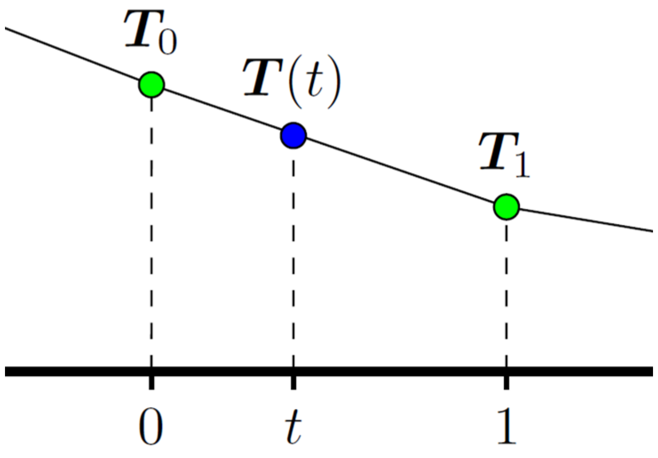
$$\boldsymbol{T}(t) = (1-t)\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1$$
$$= \boldsymbol{T}_0 - t\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1$$
$$= \boldsymbol{T}_0 + t\,(\boldsymbol{T}_1 - \boldsymbol{T}_0)$$

$$\exp\left(t\ \log(\boldsymbol{T}_0^{-1}\boldsymbol{T}_1)\right)$$



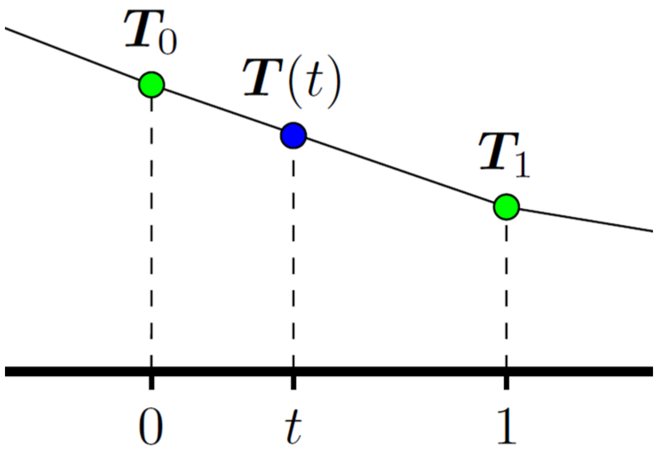
Cumulative Splines for SE3

$$\begin{aligned}\boldsymbol{T}(t) &= (1-t)\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1 \\ &= \boldsymbol{T}_0 - t\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1 \\ &= \boxed{\boldsymbol{T}_0 +}\, t\,(\boldsymbol{T}_1 - \boldsymbol{T}_0) \\ &= \boxed{\boldsymbol{T}_0}\, \exp\left(t\, \log(\boldsymbol{T}_0^{-1}\boldsymbol{T}_1)\right)\end{aligned}$$



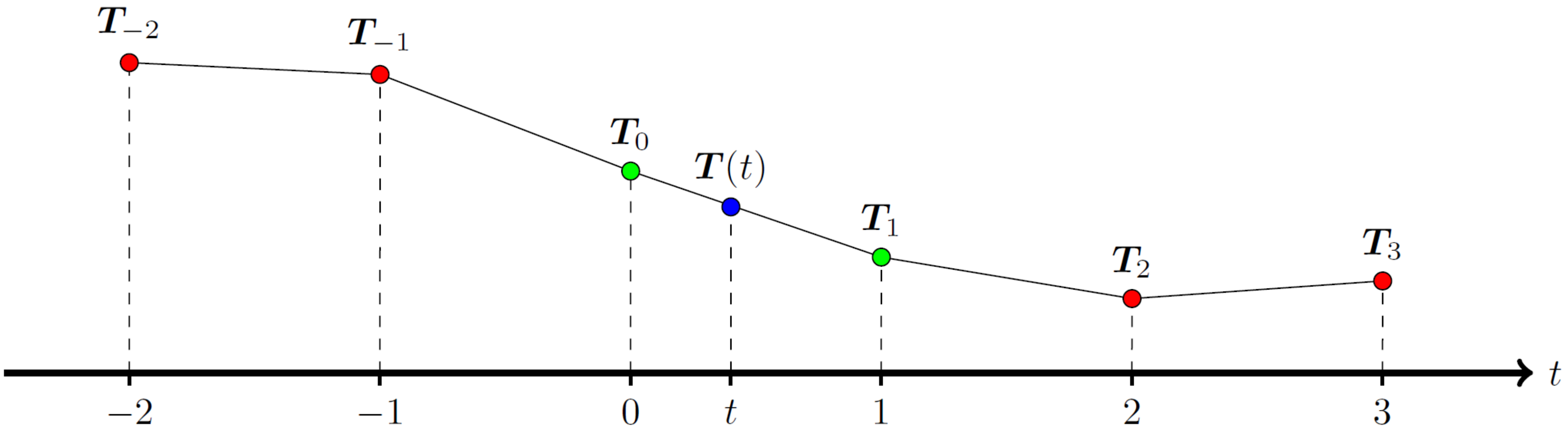
Cumulative Splines for SE3

$$
\begin{aligned}
\boldsymbol{T}(t) &= (1-t)\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1 \\
&= \boldsymbol{T}_0 - t\,\boldsymbol{T}_0 + t\,\boldsymbol{T}_1 \\
&= \boldsymbol{T}_0 + t\,(\boldsymbol{T}_1 - \boldsymbol{T}_0) \\
&= \boxed{\boldsymbol{T}_0}\exp\left(\boxed{t}\log(\boxed{\boldsymbol{T}_0^{-1}\boldsymbol{T}_1})\right) \\
&= \boxed{\boldsymbol{T}_0}\prod_{k=1}^{1}\exp\left(\boxed{\boldsymbol{B}_k\,(t)}\log(\boxed{\boldsymbol{T}_{k-1}^{-1}\,\boldsymbol{T}_k})\right)
\end{aligned}
$$

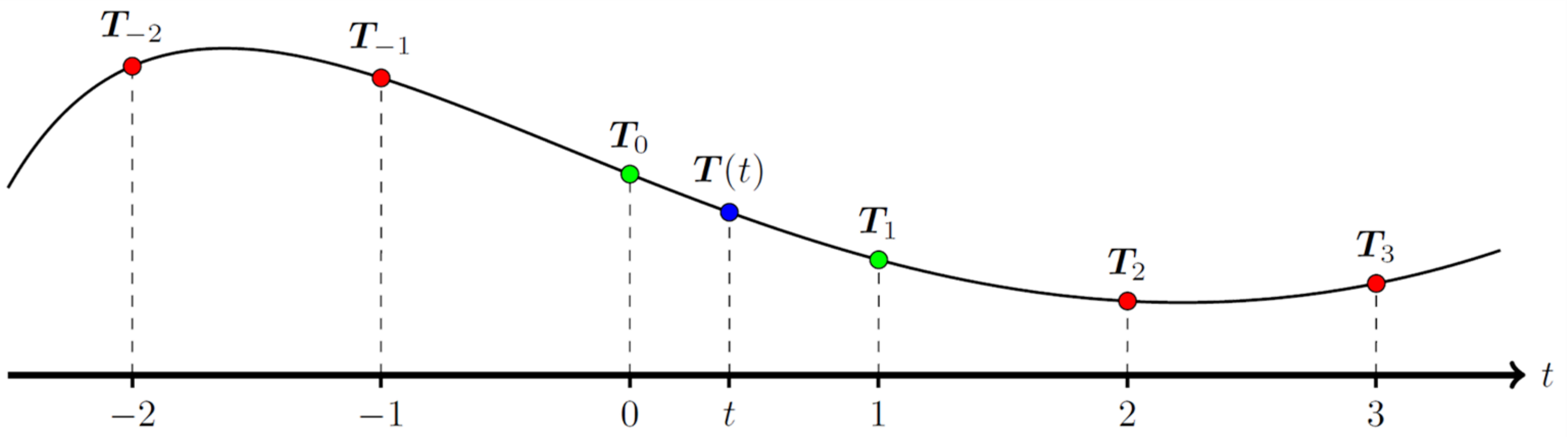


$$
\boldsymbol{B}(t) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \end{pmatrix}
$$

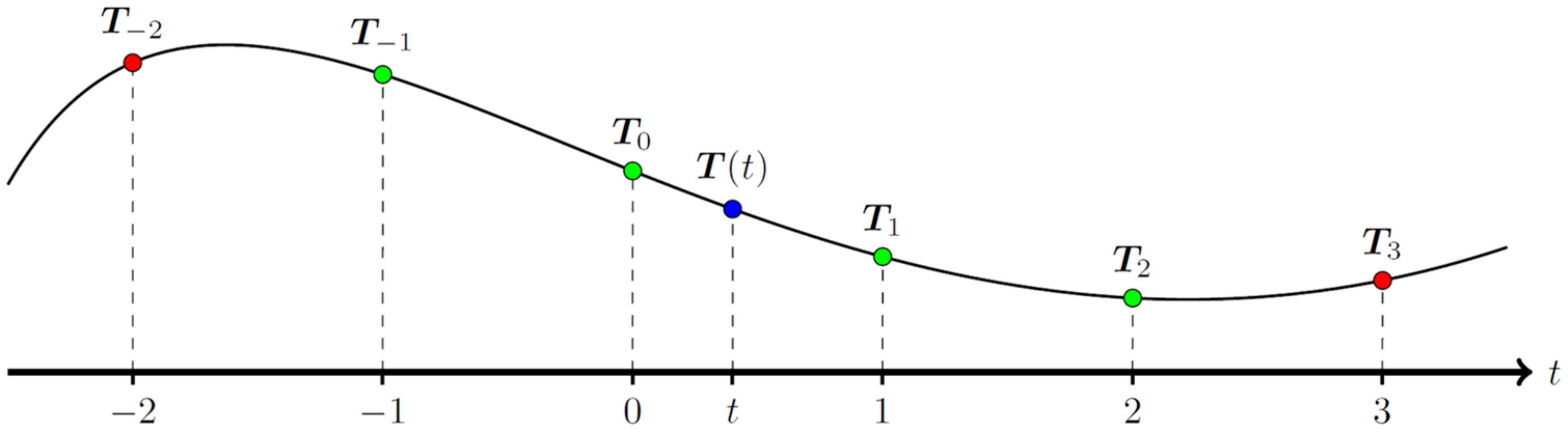
Higher Order Splines



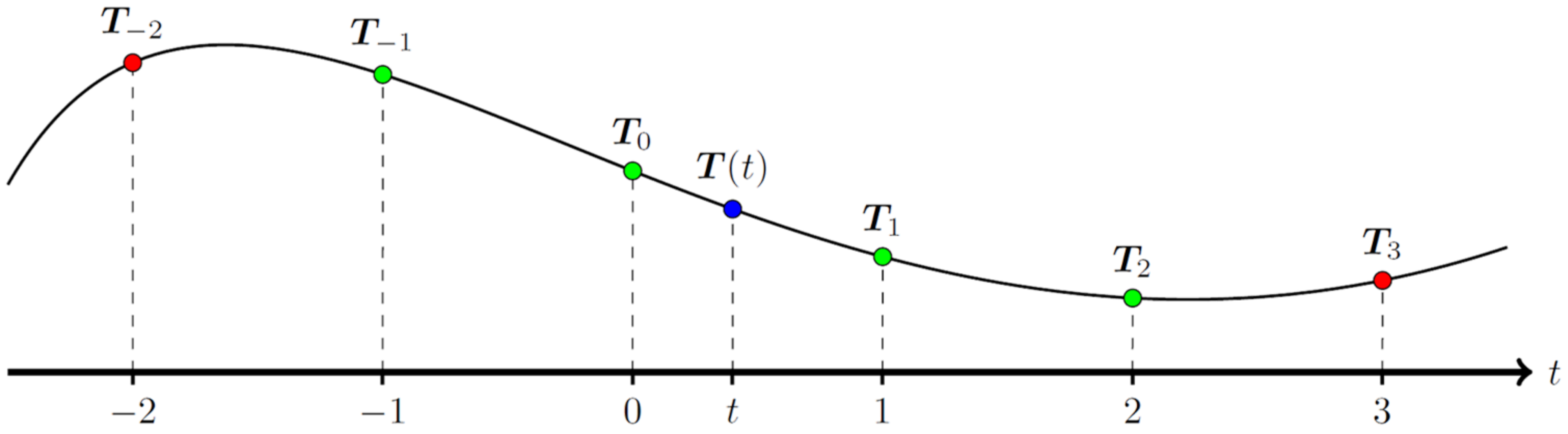
Higher Order Splines



Higher Order Splines

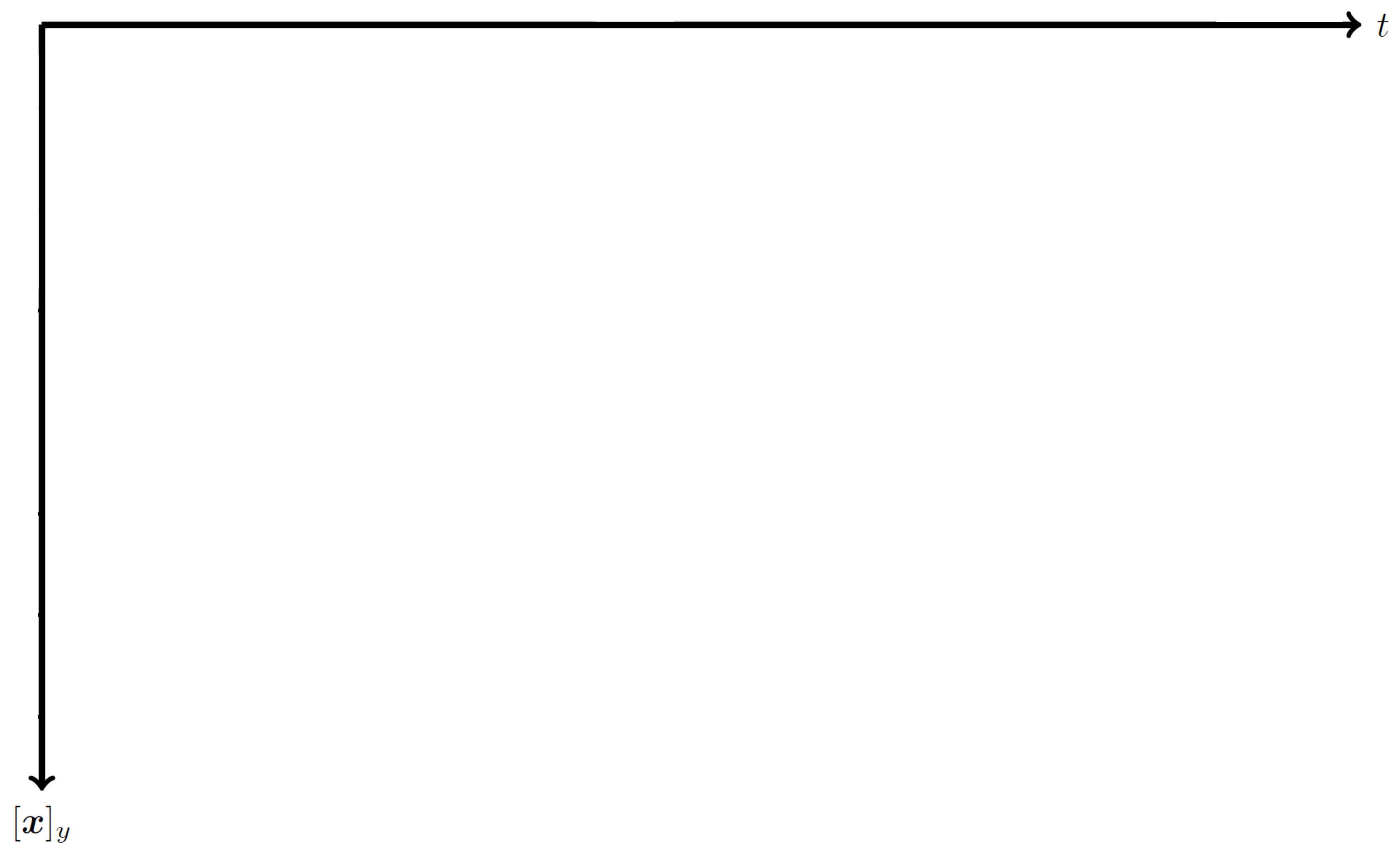


Higher Order Splines

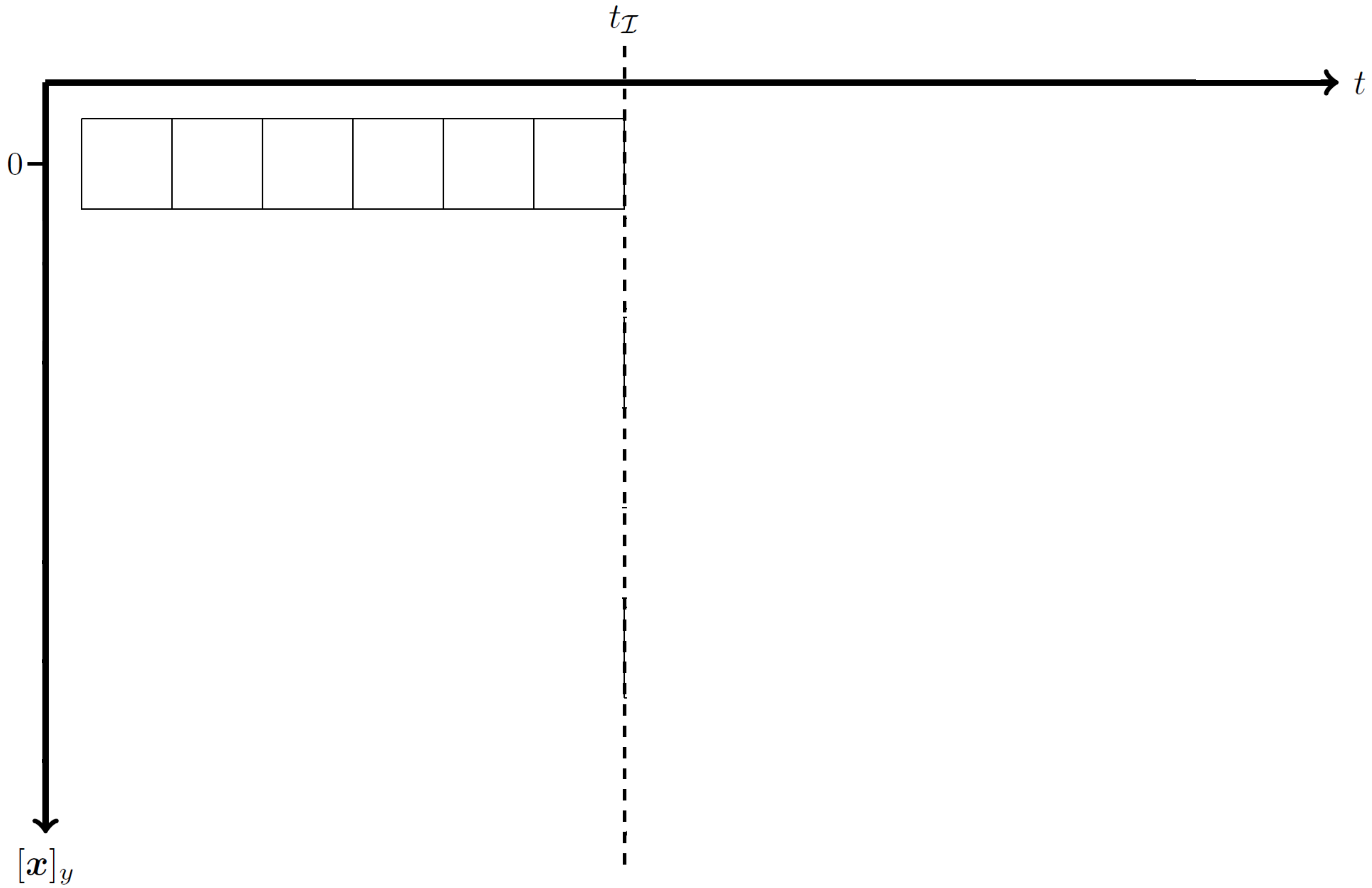


$$\boldsymbol{T}(t) = \boldsymbol{T}_{-1} \prod_{k=0}^{2} \exp\left(\boldsymbol{B}_{k+1}(t) \log(\boldsymbol{T}_{k-1}^{-1} \boldsymbol{T}_k)\right) \qquad \boldsymbol{B}(t) = \frac{1}{6} \begin{pmatrix} 6 & 0 & 0 & 0 \\ 5 & 3 & -3 & 1 \\ 1 & 3 & 3 & -2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

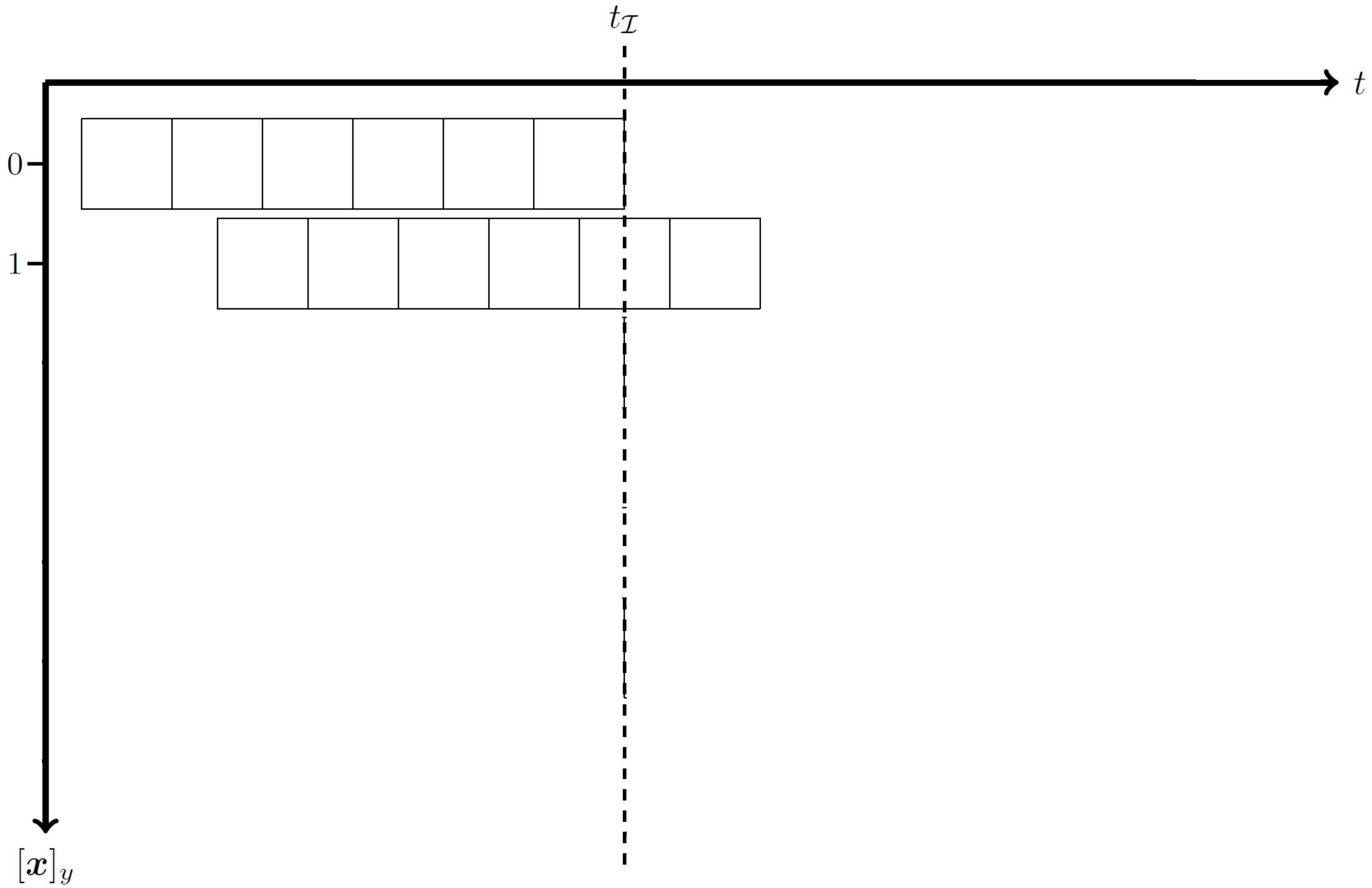
Rolling Shutter Cameras



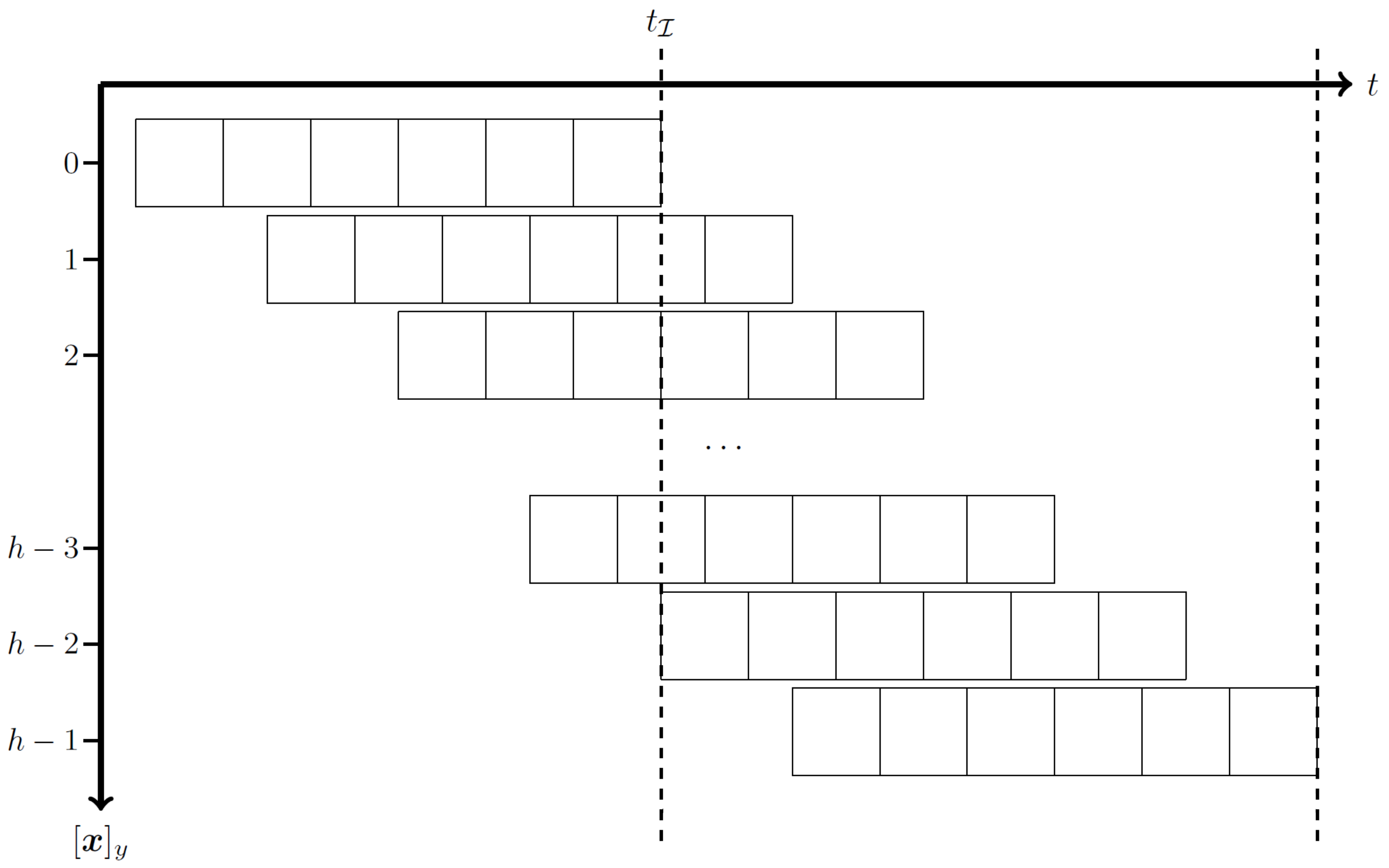
Rolling Shutter Cameras



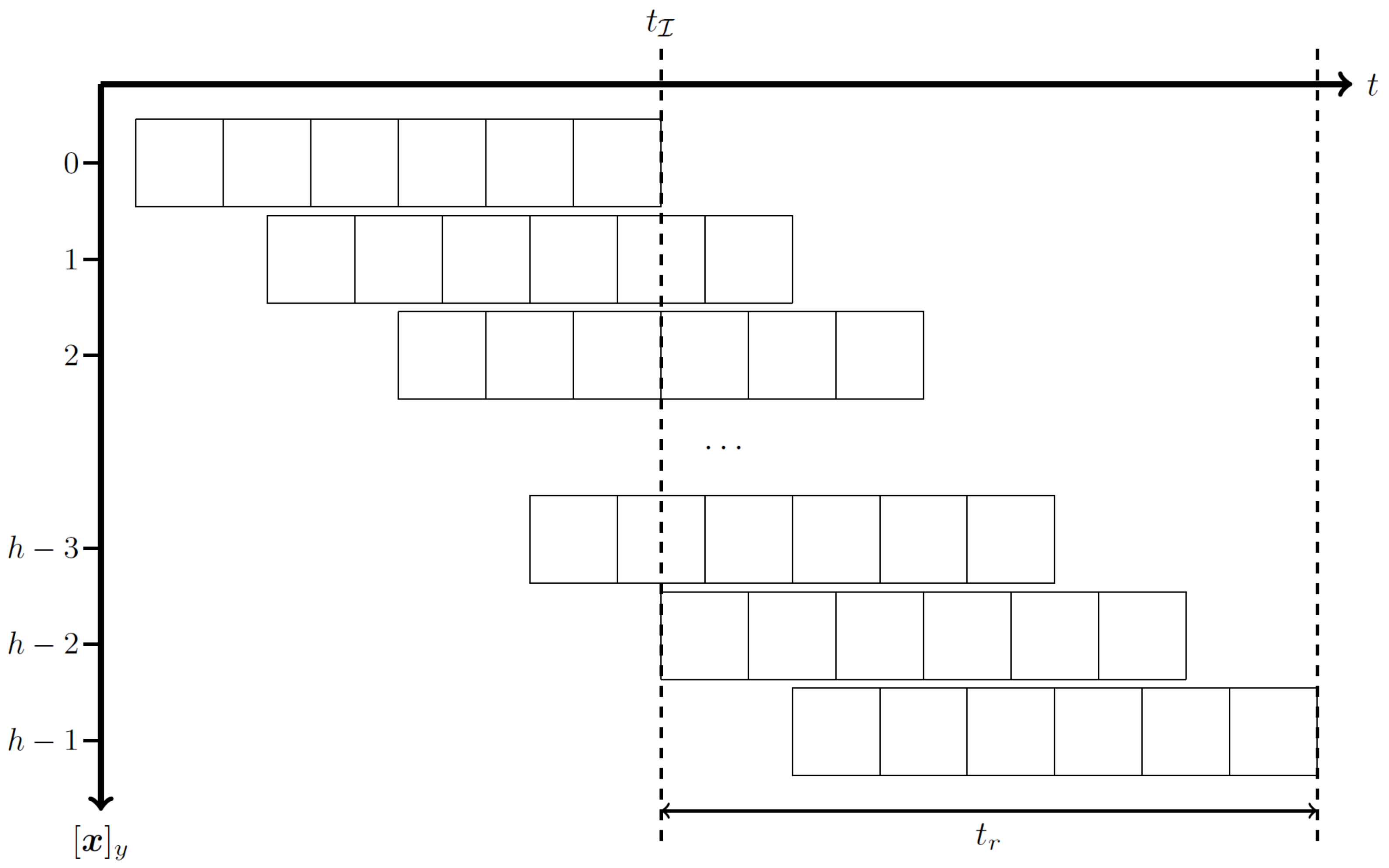
Rolling Shutter Cameras



Rolling Shutter Cameras



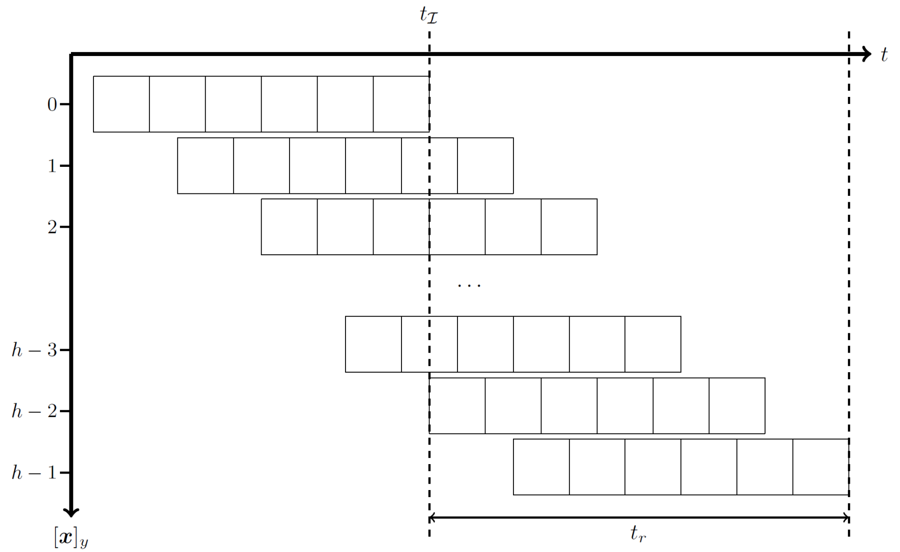
Rolling Shutter Cameras



Rolling Shutter Projection

- Pixel $\boldsymbol{x}$ captured at

$$t_{\mathcal{I}} + t_x = t_{\mathcal{I}} + {}^{t_r}/_{h}[\boldsymbol{x}]_y$$



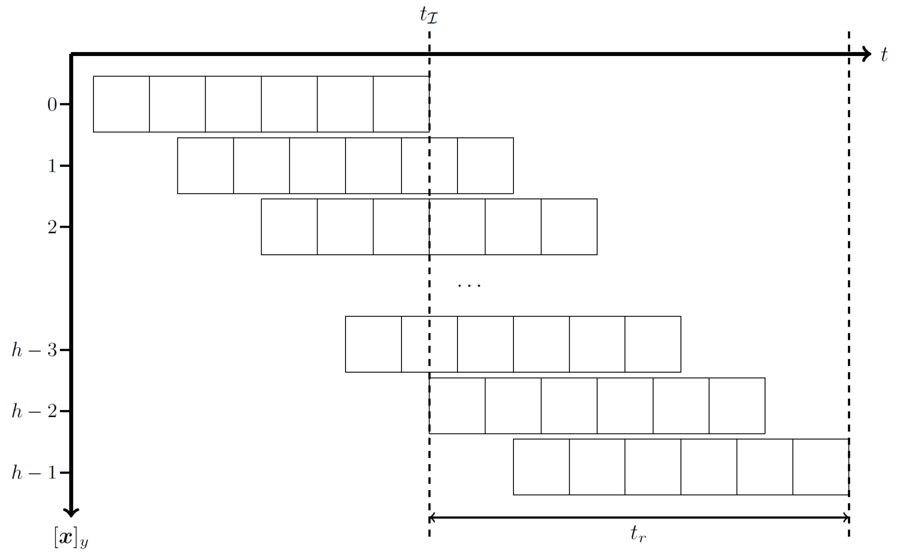
Rolling Shutter Projection

- Pixel $\boldsymbol{x}$ captured at

$$t_{\mathcal{I}} + t_x = t_{\mathcal{I}} + {}^{t_r}\!/_{h}[\boldsymbol{x}]_y$$

- Projection constraint

$$ {}^{t_r}\!/_{h}\left[\pi(\boldsymbol{T}(t_{\mathcal{I}} + t_x)\,\boldsymbol{p})\right]_y \overset{!}{=} t_x$$



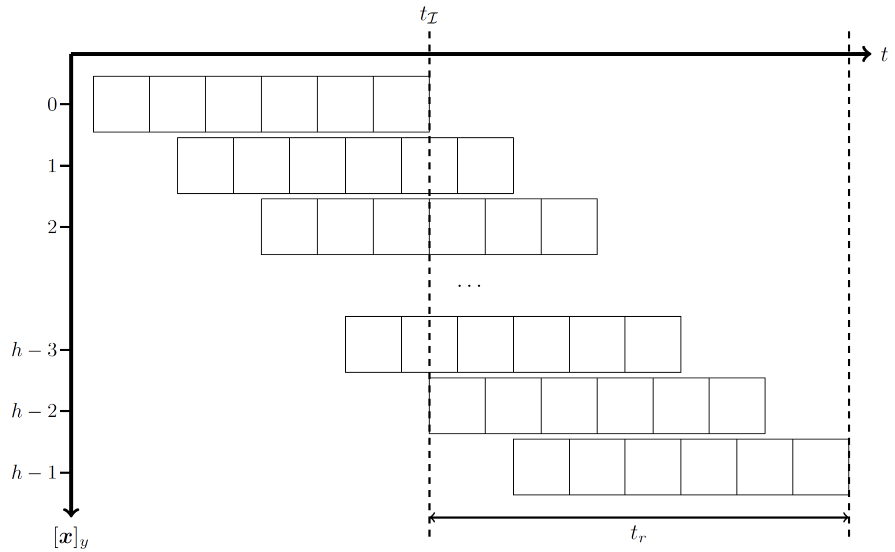
Rolling Shutter Projection

- Pixel $\boldsymbol{x}$ captured at

$$t_{\mathcal{I}} + t_x = t_{\mathcal{I}} + {}^{t_r}/_{h}[\boldsymbol{x}]_y$$

- Projection constraint

$${}^{t_r}/_{h}\left[\pi(\boldsymbol{T}(t_{\mathcal{I}} + \boxed{t_x})\,\boldsymbol{p})\right]_y \stackrel{!}{=} \boxed{t_x}$$



Rolling Shutter Projection

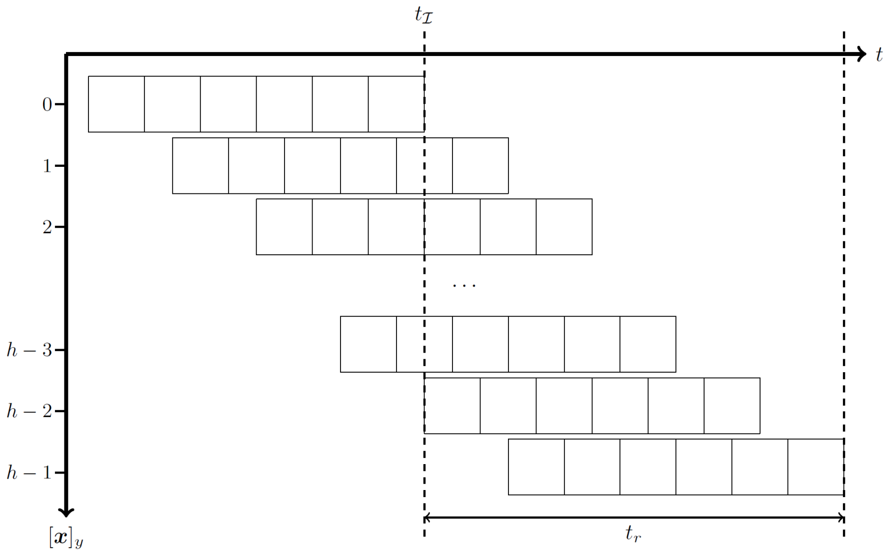
- Pixel $\boldsymbol{x}$ captured at

$$t_{\mathcal{I}} + t_x = t_{\mathcal{I}} + {}^{t_r}\!/_{h}[\boldsymbol{x}]_y$$

- Projection constraint

$${}^{t_r}\!/_{h}\left[\pi\left(\boldsymbol{T}\left(t_{\mathcal{I}} + \boxed{t_x}\right)\boldsymbol{p}\right)\right]_y \stackrel{!}{=} \boxed{t_x}$$

- Solve numerically



Rolling Shutter RGB-D Cameras

- Two independent RS cameras
  - Unsynchronized
  - Different read out times

- Assumptions
  - Calibrated intrinsics / extrinsics / read out times
  - RGB + depth **not** registered

Dense Tracking and Mapping

- Goal: Estimate $\boldsymbol{T}(t)$ to build a map without RS artifacts

- Minimize pixel-wise error w.r.t. control poses
  - Geometric error
  - Photometric error

- Update map with rectified images once $\boldsymbol{T}(t)$ converged

Dense Image Alignment

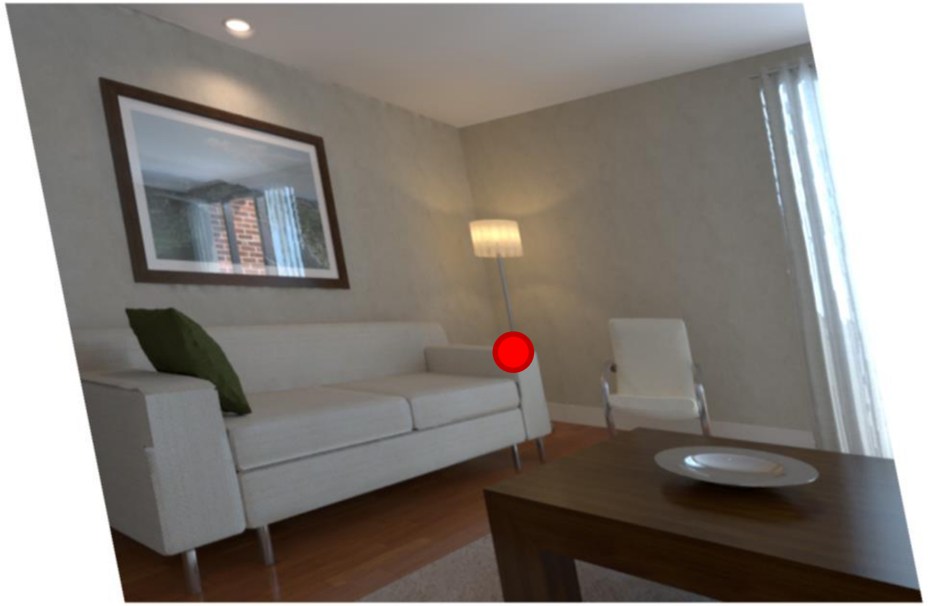
reference $\mathcal{I}_{\text{ref}}$ / $\mathcal{Z}_{\text{ref}}$

current $\mathcal{I}_{\text{cur}}$ / $\mathcal{Z}_{\text{cur}}$

Geometric Error

reference $\mathcal{Z}_{\mathrm{ref}}$ @ $t_{\mathcal{Z},\mathrm{ref}}$

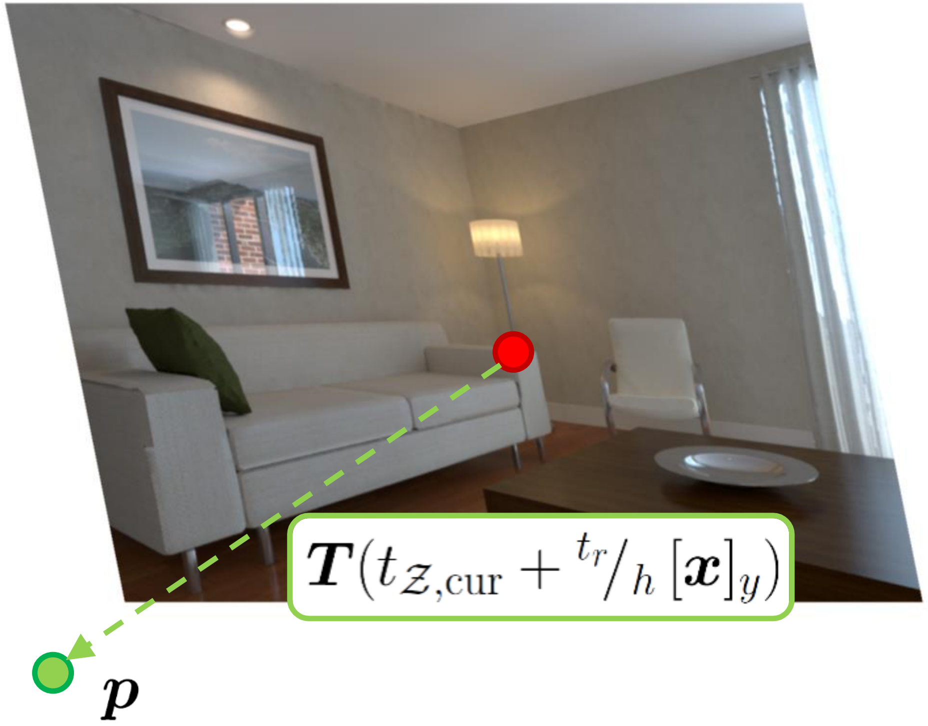
current $\mathcal{Z}_{\mathrm{cur}}$ @ $t_{\mathcal{Z},\mathrm{cur}}$



Geometric Error

reference $\mathcal{Z}_{\text{ref}}$ @ $t_{\mathcal{Z},\text{ref}}$

current $\mathcal{Z}_{\text{cur}}$ @ $t_{\mathcal{Z},\text{cur}}$

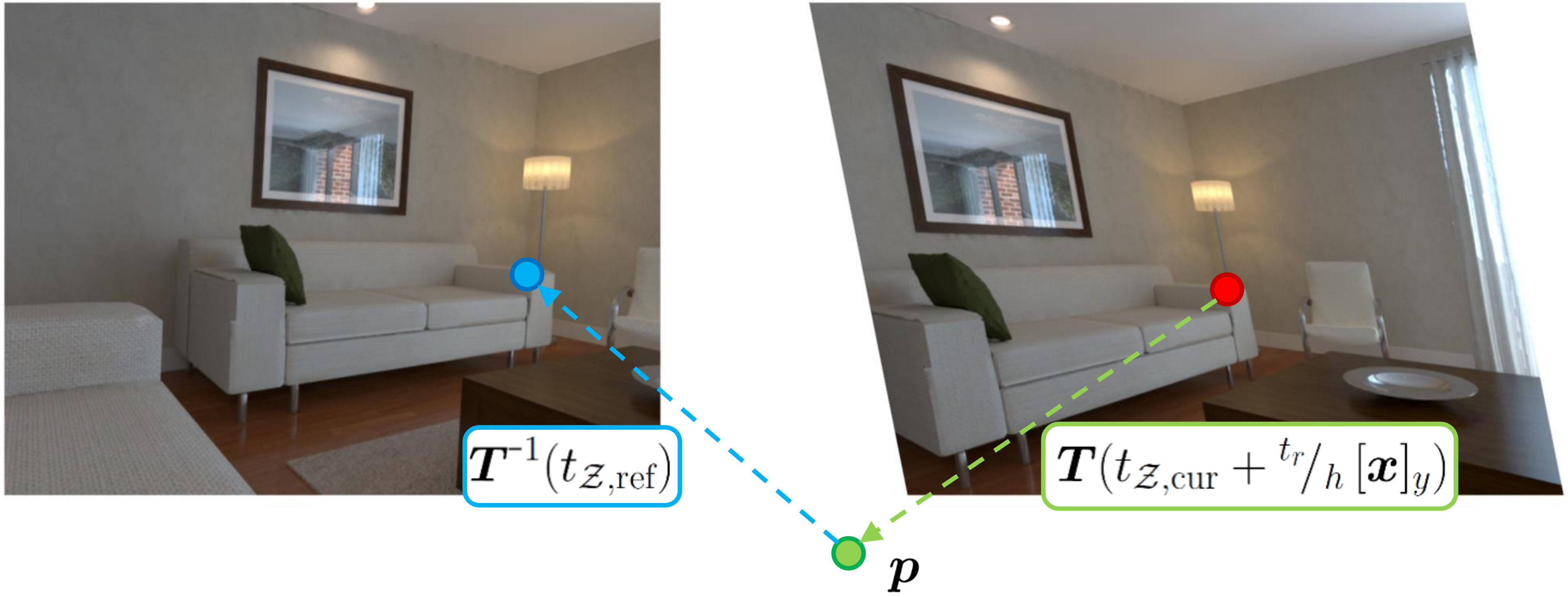


$$\boldsymbol{p} = \boldsymbol{T}(t_{\mathcal{Z},\text{cur}} + {}^{t_r}\!/_{h}\,[\boldsymbol{x}]_y)\,\pi^{-1}(\boldsymbol{x}, \mathcal{Z}_{\text{cur}}(\boldsymbol{x}))$$

Geometric Error

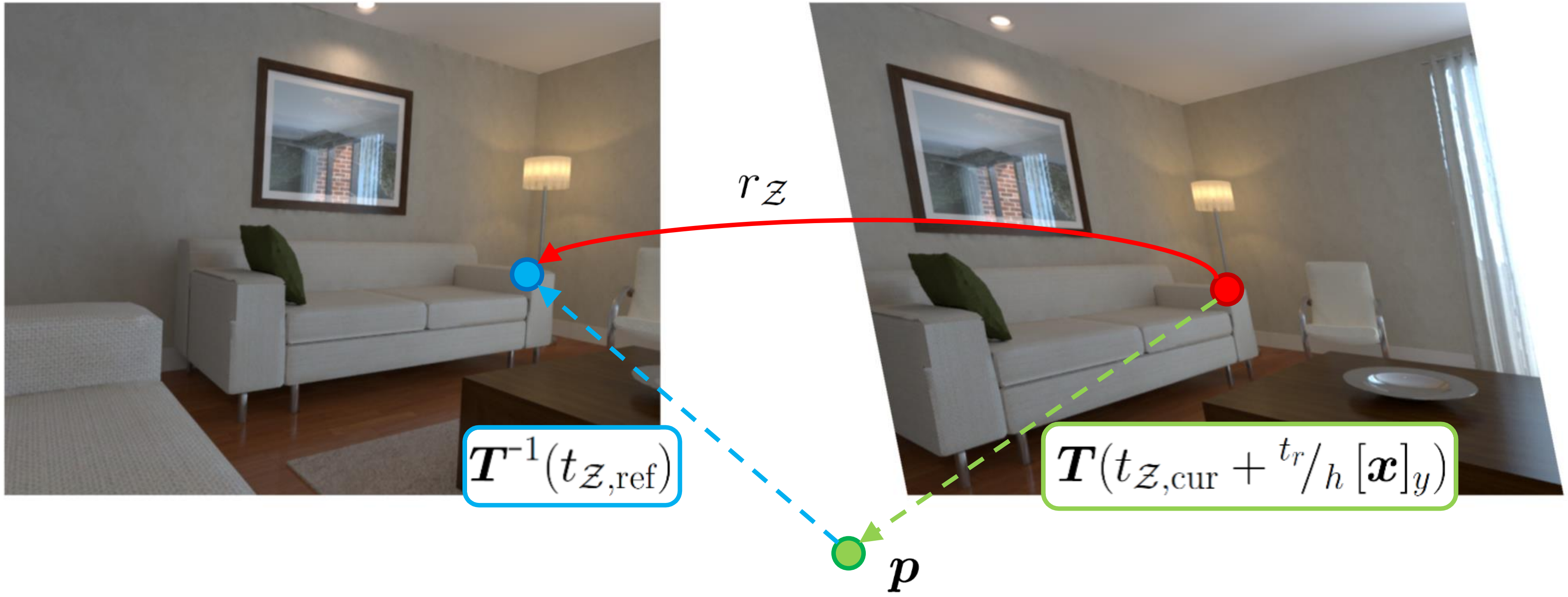
reference $\mathcal{Z}_{\text{ref}}$ $@\, t_{\mathcal{Z},\text{ref}}$

current $\mathcal{Z}_{\text{cur}}$ $@\, t_{\mathcal{Z},\text{cur}}$



Geometric Error

reference $\mathcal{Z}_{\text{ref}}$ @ $t_{\mathcal{Z},\text{ref}}$ current $\mathcal{Z}_{\text{cur}}$ @ $t_{\mathcal{Z},\text{cur}}$



$$r_{\mathcal{Z}} = \mathcal{Z}_{\text{ref}}(\pi(\boldsymbol{T}^{-1}(t_{\mathcal{Z},\text{ref}})\,\boldsymbol{p})) - [\boldsymbol{T}^{-1}(t_{\mathcal{Z},\text{ref}})\,\boldsymbol{p}]_Z$$

Photometric Error

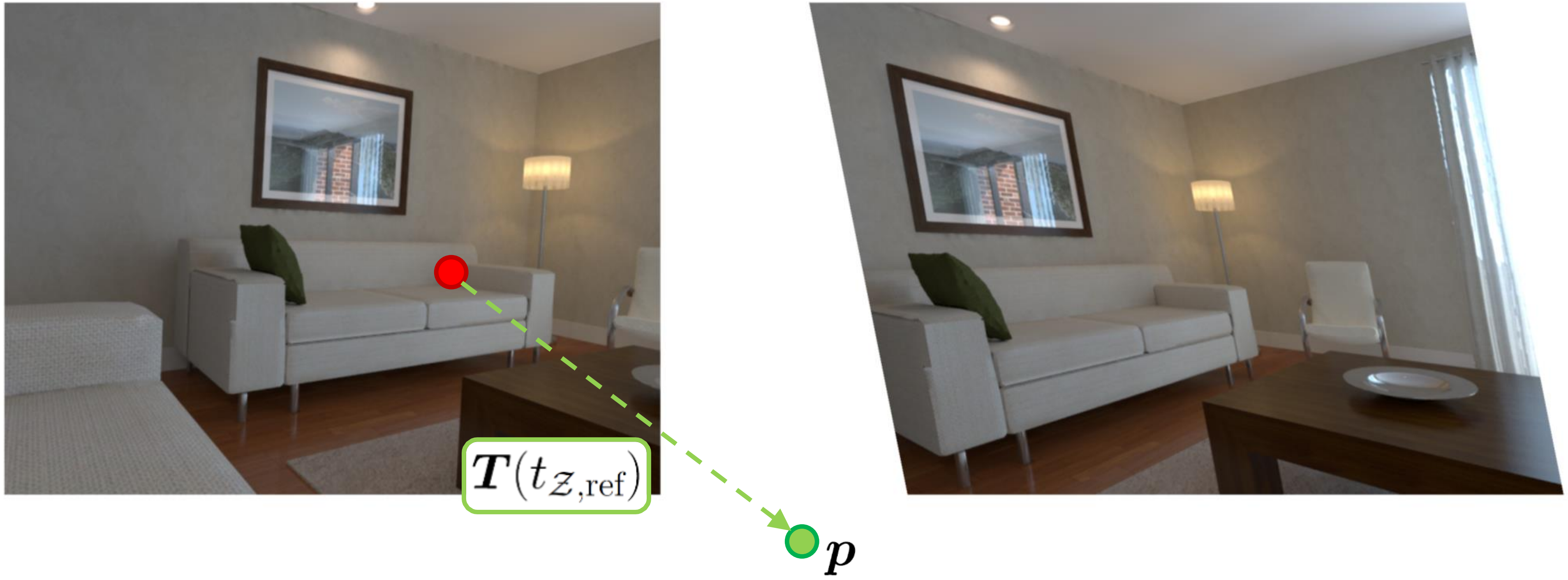
reference $\mathcal{I}_{\text{ref}}$ $@\, t_{\mathcal{Z},\text{ref}}$

current $\mathcal{I}_{\text{cur}}$ $@\, t_{\mathcal{I},\text{cur}}$

Photometric Error

reference $\mathcal{I}_{\text{ref}}$ $@\, t_{\mathcal{Z},\text{ref}}$

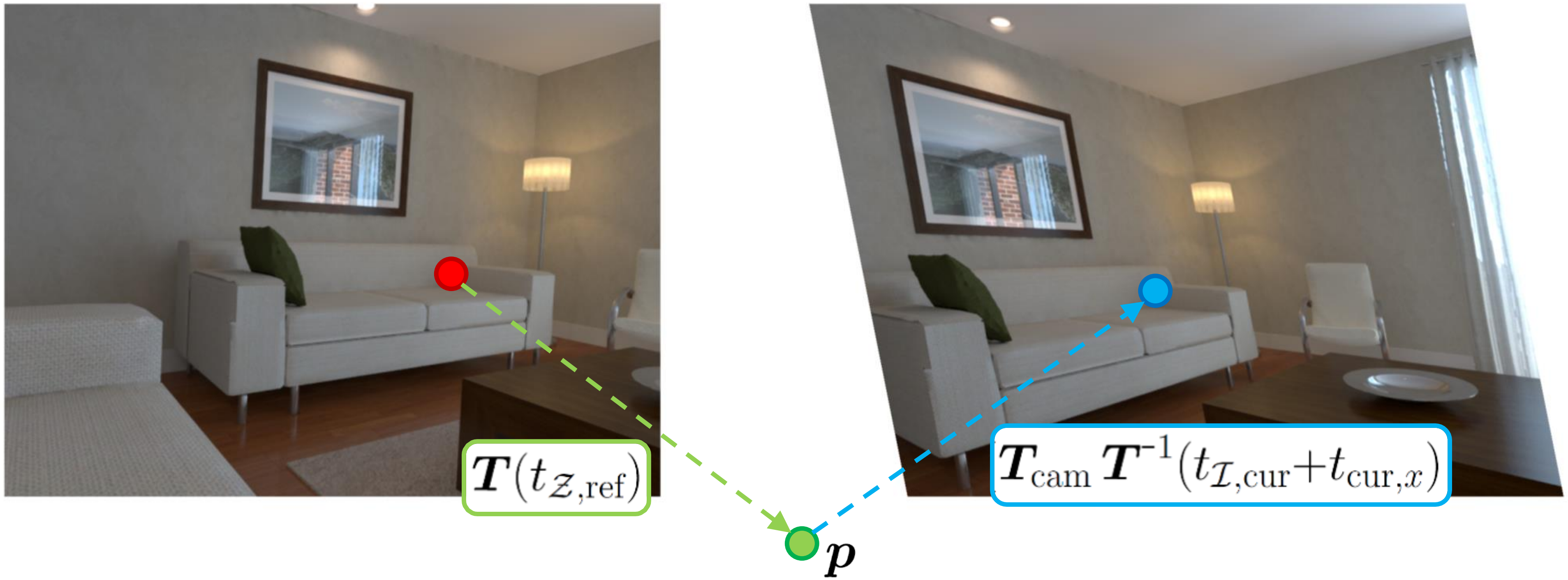
current $\mathcal{I}_{\text{cur}}$ $@\, t_{\mathcal{I},\text{cur}}$



$$\boldsymbol{p} = \boldsymbol{T}(t_{\mathcal{Z},\text{ref}})\, \pi^{-1}(\boldsymbol{x}, \mathcal{Z}_{\text{ref}}(\boldsymbol{x}))$$

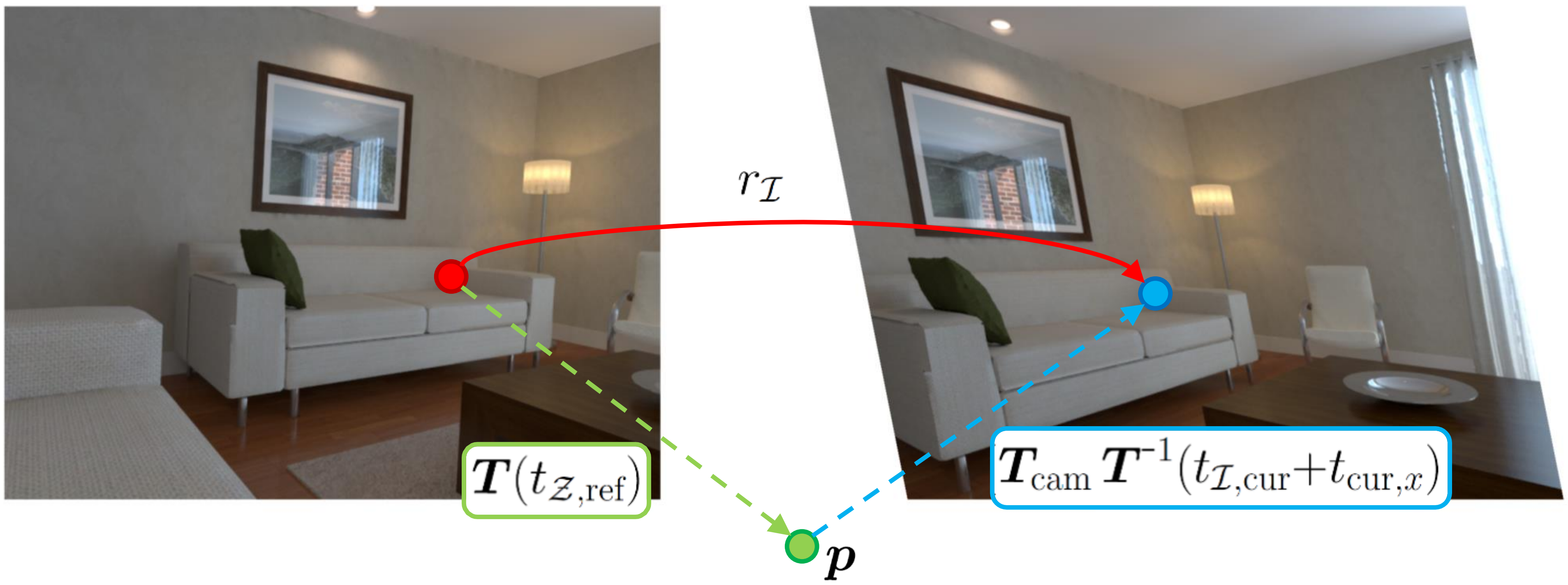
Photometric Error

reference $\mathcal{I}_{\text{ref}}$ $@\, t_{\mathcal{Z},\text{ref}}$ current $\mathcal{I}_{\text{cur}}$ $@\, t_{\mathcal{I},\text{cur}}$



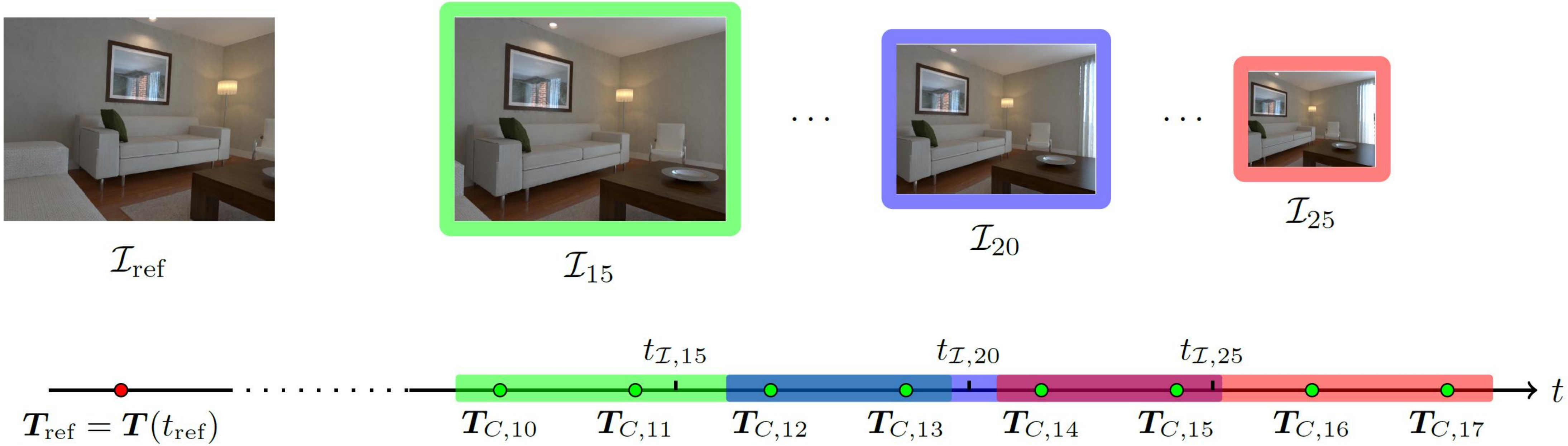
Photometric Error

reference $\mathcal{I}_{\text{ref}}$ @ $t_{\mathcal{Z},\text{ref}}$ current $\mathcal{I}_{\text{cur}}$ @ $t_{\mathcal{I},\text{cur}}$



$$r_{\mathcal{I}} = \mathcal{I}_{\text{cur}}(\pi(\boldsymbol{T}_{\text{cam}}\,\boldsymbol{T}^{-1}(t_{\mathcal{I},\text{cur}}+t_{\text{cur},x})\,\boldsymbol{p})) - \mathcal{I}_{\text{ref}}(\boldsymbol{x})$$

Optimization

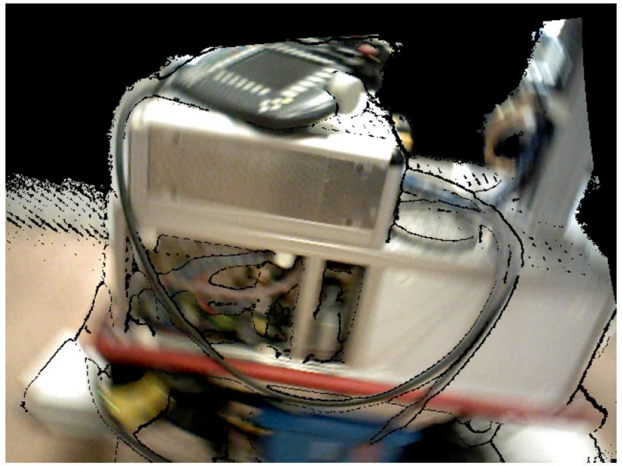


- Non-linear least squares optimized with Gauss-Newton
- Coarse-to-fine + robust weights

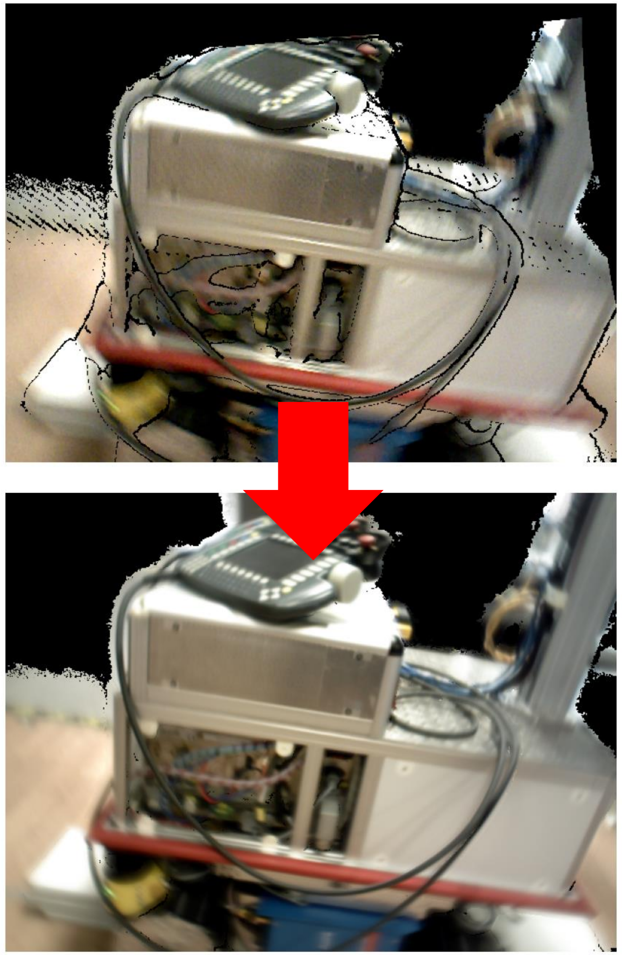
Mapping with Keyframes

- Map is set of keyframes
- Latest keyframe acts as reference image
- Frames with converged control points warped to keyframe
- Fusion of RGB and depth using weighted average

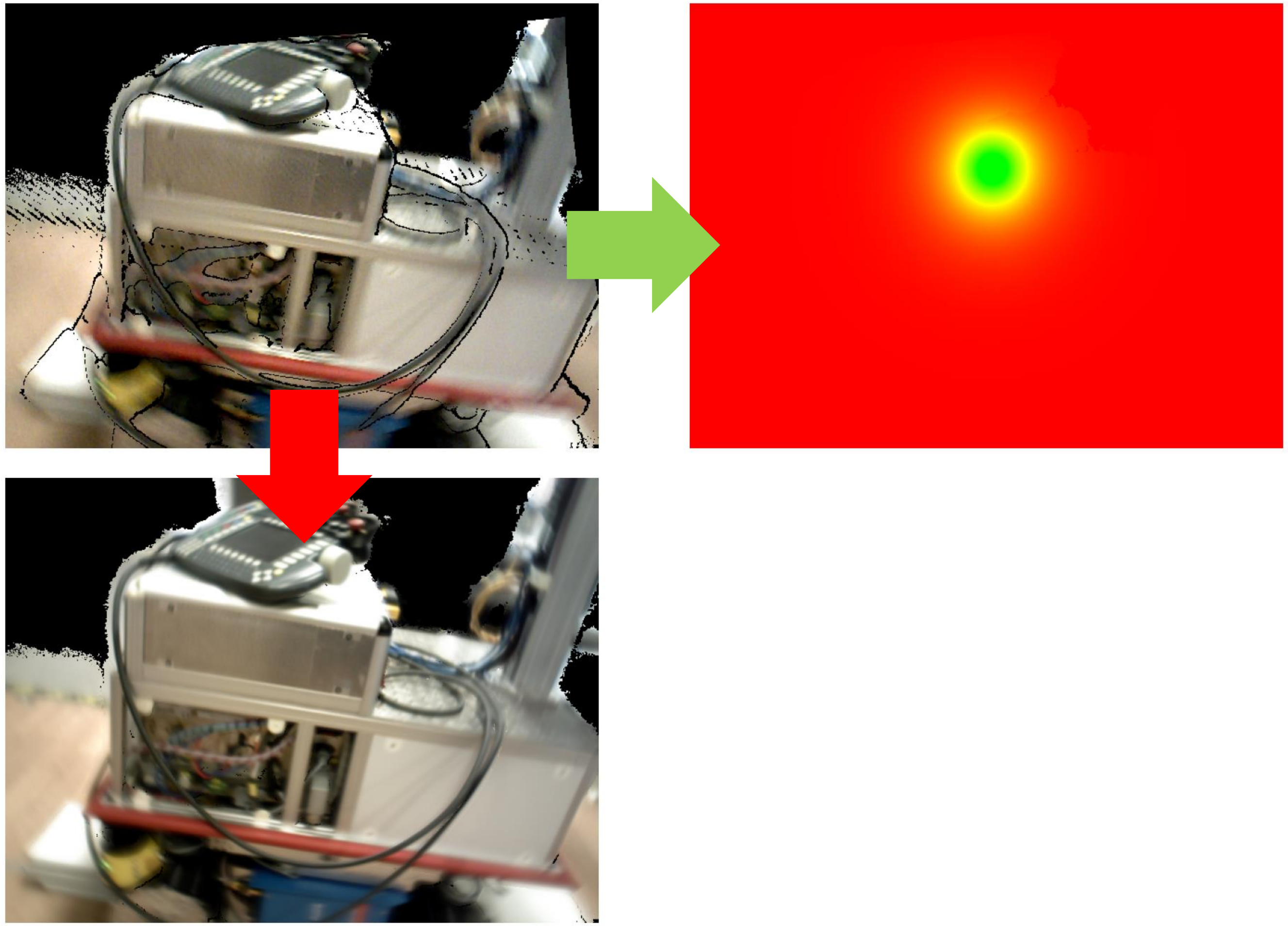
Blur-aware Weights for RGB Fusion



Blur-aware Weights for RGB Fusion

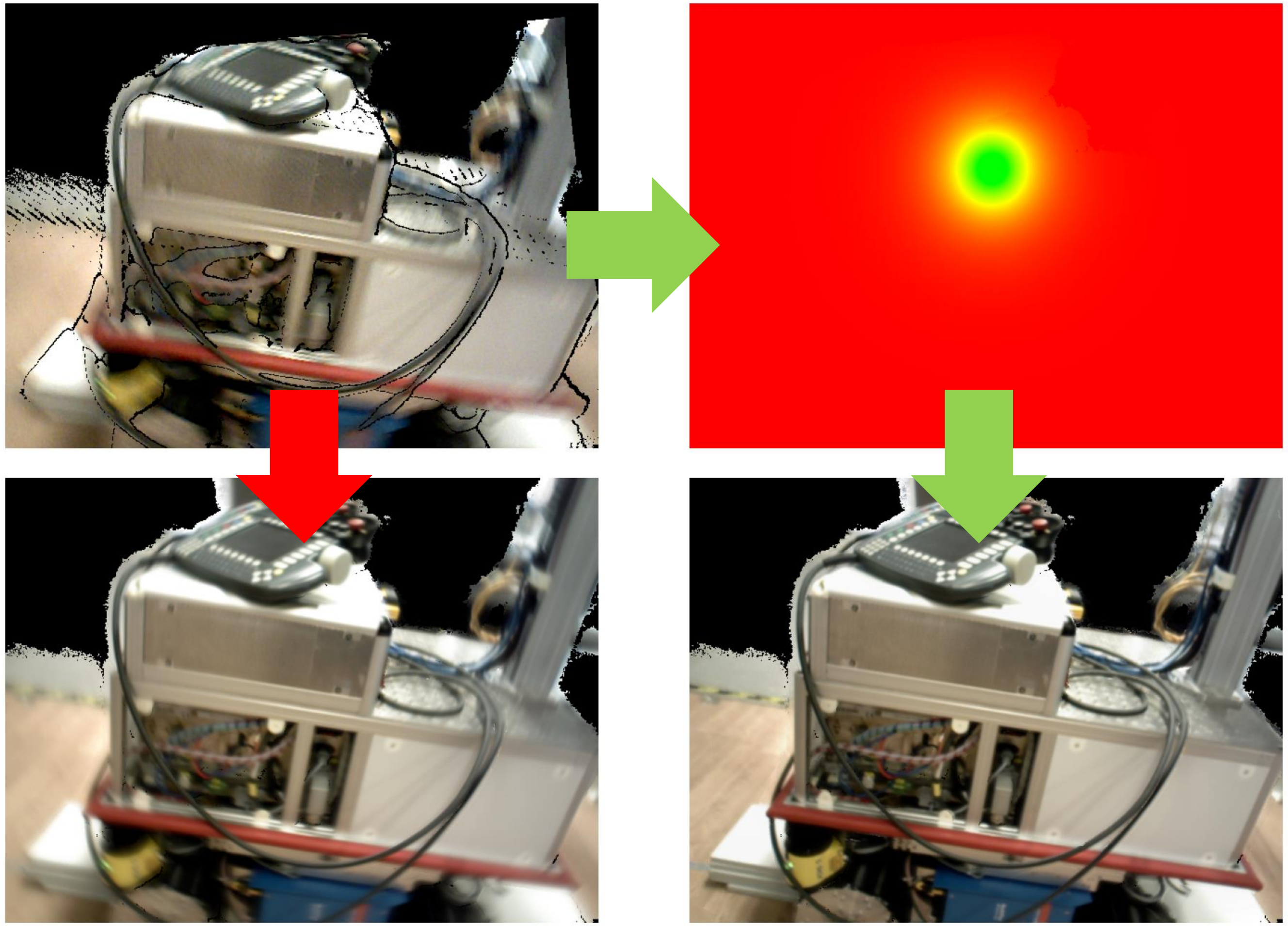


Blur-aware Weights for RGB Fusion



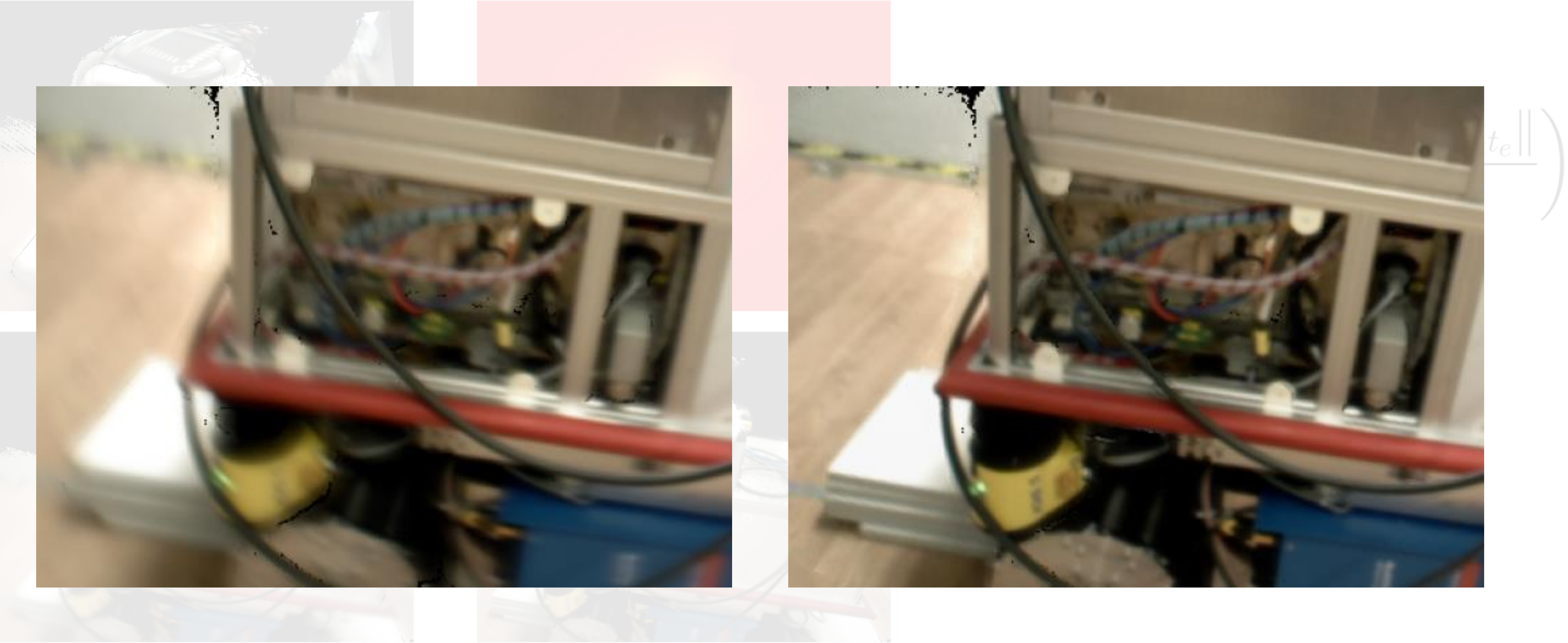
$$w_{\mathcal{C}}(\boldsymbol{p}) = \exp\left(-\frac{\|\boldsymbol{x}_t - \boldsymbol{x}_{t-t_e}\|}{\sigma^2}\right)$$

Blur-aware Weights for RGB Fusion



$$w_{\mathcal{C}}(\boldsymbol{p}) = \exp\left(-\frac{\|\boldsymbol{x}_t - \boldsymbol{x}_{t-t_e}\|}{\sigma^2}\right)$$

Blur-aware Weights for RGB Fusion



Evaluation

- Synthetic: ICL-NUIM living room scene
  - Rendered 4 trajectories each with GS and RS
  - No motion blur

Evaluation

- Synthetic: ICL-NUIM living room scene
  - Rendered 4 trajectories each with GS and RS
  - No motion blur
- Real: PrimeSense Carmine within MoCap
  - 6 sequences with groundtruth trajectory
  - Higher translational and rotational velocities than TUM RGB-D benchmark

Evaluation

- Synthetic: ICL-NUIM living room scene
  - Rendered 4 trajectories each with GS and RS
  - No motion blur
- Real: PrimeSense Carmine within MoCap
  - 6 sequences with groundtruth trajectory
  - Higher translational and rotational velocities than TUM RGB-D benchmark
  - Existing benchmarks not usable due to pre-registered depth images

=> Please provide raw sensor data in future datasets! <=

Results

- Baseline: Geometric Alignment to KF with GS and fusion

Results

- Baseline: Geometric Alignment to KF with GS and fusion

	Translation Drift	Rotational Drift
Spline + GS + G	59.0%	27.3%
Spline + GS + G + P	60.9%	27.6%
Spline + RS + G	68.8%	61.0%
Spline + RS + G + P	71.5%	61.3%

Results


Dense Continuous-Time Tracking and Mapping
with Rolling Shutter RGB-D Cameras
Christian Kerl, Jörg Stückler, Daniel Cremers
Computer Vision and Pattern Recognition Group
Department of Computer Science
Technische Universität München

Conclusion

- Dense tracking method to estimate continuous-time trajectory
- Allows to model/remove RS
- Blur-aware weights to suppress motion blur artifacts
- Spline + RS model improve accuracy considerably
- Complementary to existing mapping systems