DTAM: Dense Tracking and Mapping in Real-Time

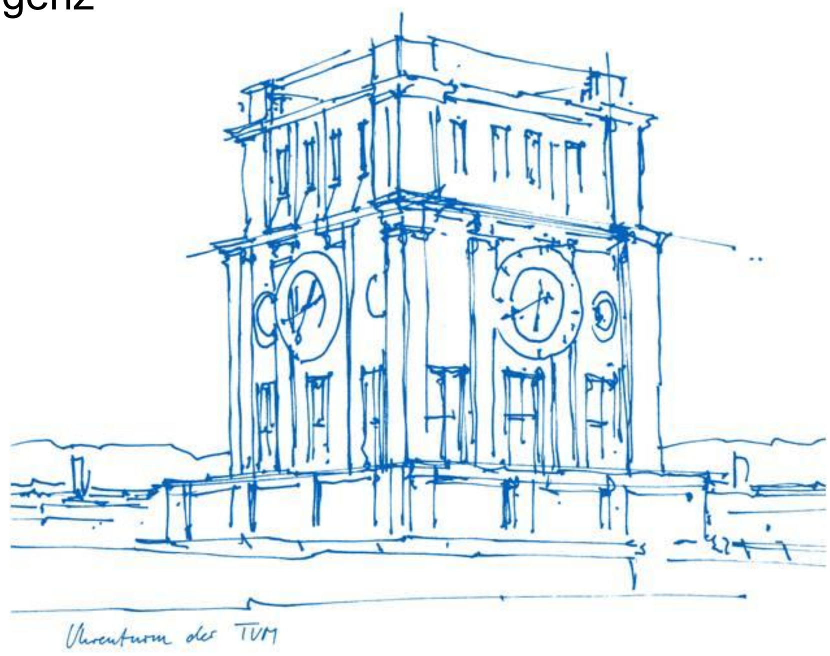
Chenguang Huang

Technische Universität München

Fakultät für Informatik

Lehrstuhl für Bildverarbeitung und Künstliche Intelligenz

20.11.2019



Introduction

monocular SLAM

- use only one camera
- create 3D model of the environment
- track camera pose

previous methods: PTAM

- use CPU only
- no prior model of the environment needed
- real-time
- unstable with vibration
- sparse feature

Introduction

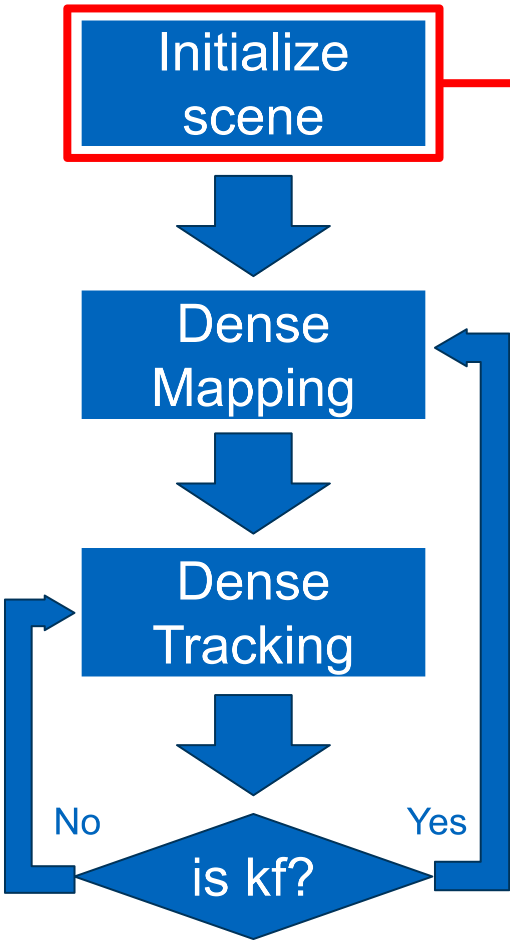
Motivation

- real-time
- track camera robustly
- create dense map

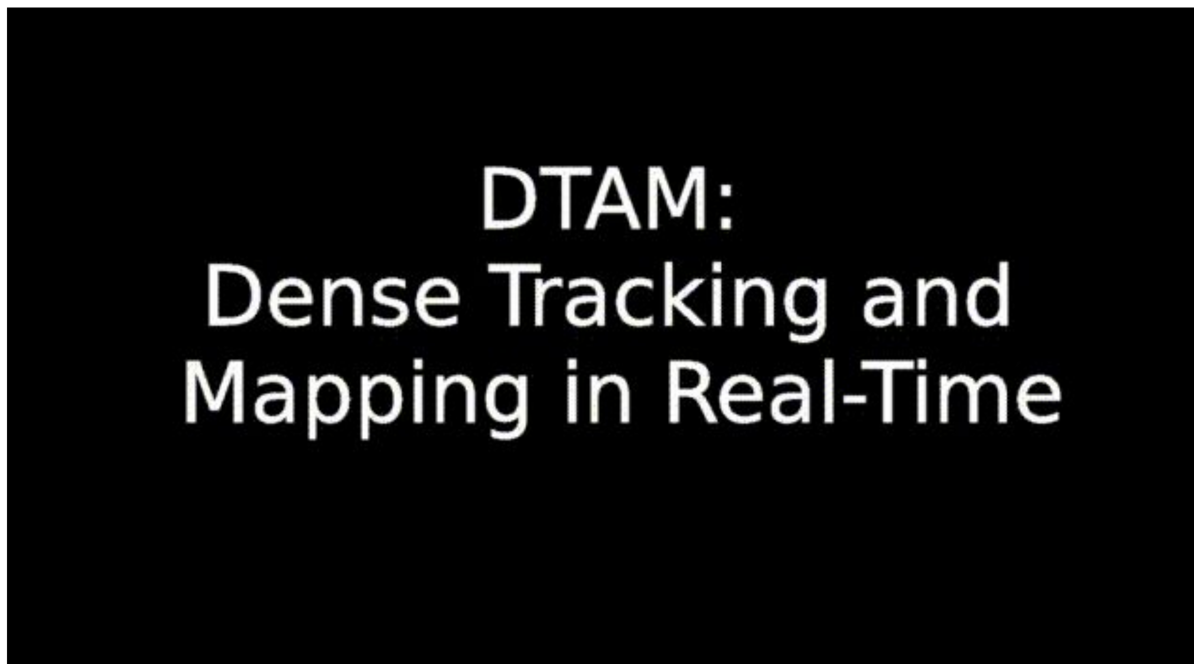
DTAM

- use GPU
- no prior model of the environment needed
- real-time
- stable with vibration and defocus
- dense map

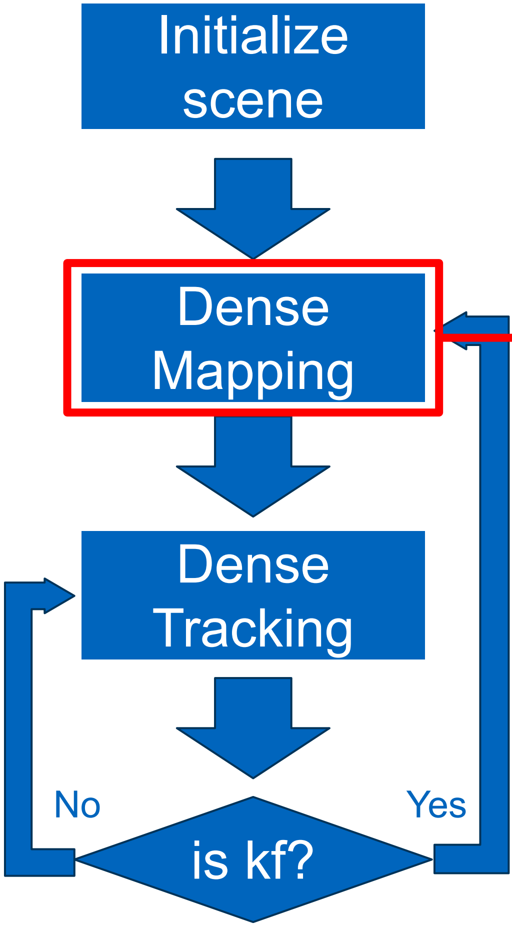
Method: Pipeline and Initialization



point feature based stereo method



Method: Dense Mapping



solve an optimization problem to find the inverse depth map

$$E_{\boldsymbol{\xi}} = \int_{\Omega} \left\{ g(\mathbf{u}) \|\nabla \boldsymbol{\xi}(\mathbf{u})\|_{\epsilon} + \lambda \mathbf{C}\left(\mathbf{u}, \boldsymbol{\xi}(\mathbf{u})\right) \right\} d\mathbf{u}$$

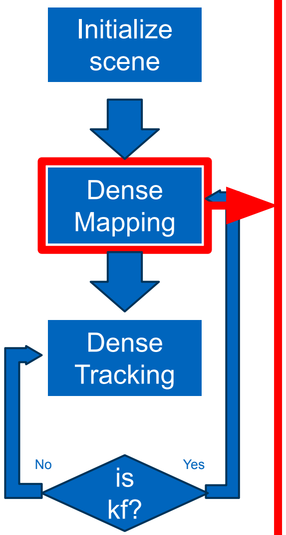
regularizer

smooth depth map but also keep sharp edge

cost volume

minimize the photometric loss

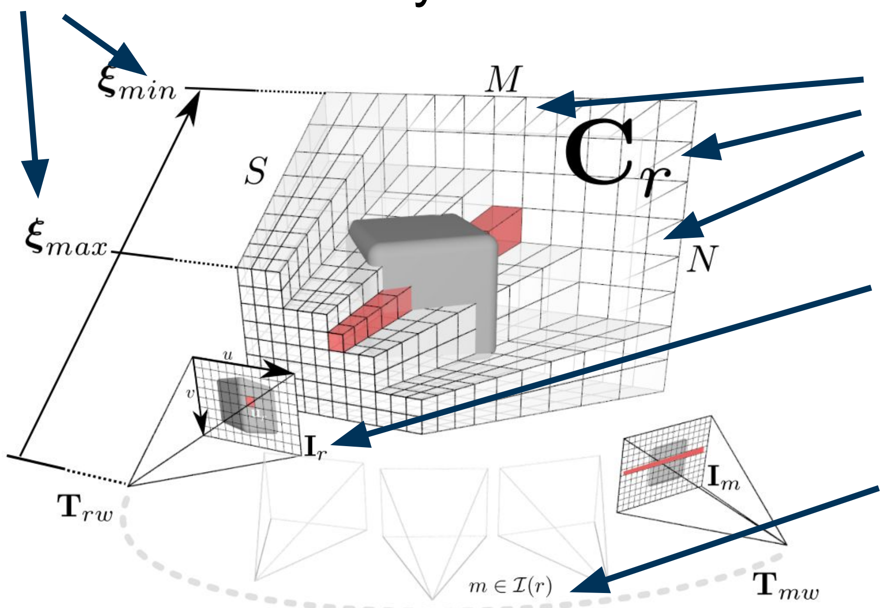
Method: Dense Mapping



$$E_{\boldsymbol{\xi}} = \int_\Omega \left\{ g(\mathbf{u}) \|\nabla \boldsymbol{\xi}(\mathbf{u})\|_\epsilon + \lambda \mathbf{C}\left(\mathbf{u}, \boldsymbol{\xi}(\mathbf{u})\right) \right\} d\mathbf{u}$$

cost volume

given inverse depth range
discretize into S layers



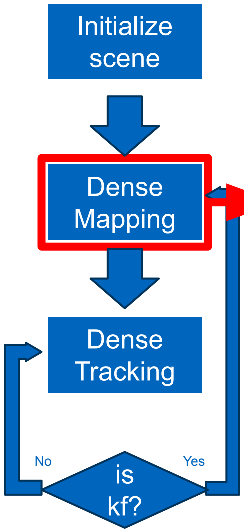
cost at each volume

reference frame r

frames that are near r

$$\mathbf{C}_r(\mathbf{u}, d) = \frac{1}{|\mathcal{I}(r)|} \sum_{m \in \mathcal{I}(r)} \|\rho_r\left(\mathbf{I}_m, \mathbf{u}, d\right)\|_1$$

Method: Dense Mapping



$$E_{\boldsymbol{\xi}} = \int_\Omega \left\{ g(\mathbf{u}) \|\nabla \boldsymbol{\xi}(\mathbf{u})\|_\epsilon + \lambda \mathbf{C}\left(\mathbf{u}, \boldsymbol{\xi}(\mathbf{u})\right) \right\} d\mathbf{u}$$

cost volume

$$\mathbf{C}_r(\mathbf{u},d) = \frac{1}{|\mathcal{I}(r)|}\sum_{m\in\mathcal{I}(r)} \left\|\rho_r\left(\mathbf{I}_m,\mathbf{u},d\right)\right\|_1$$

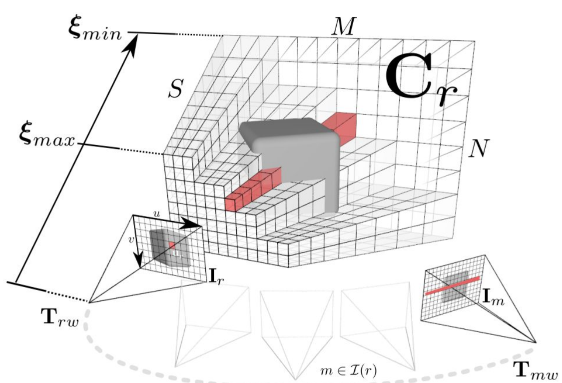
photometric loss

$$\rho_r\left(\mathbf{I}_m,\mathbf{u},d\right) = \mathbf{I}_r\left(\mathbf{u}\right) - \mathbf{I}_m\left(\pi\left(\mathrm{K}\mathrm{T}_{mr}\pi^{-1}\left(\mathbf{u},d\right)\right)\right)$$

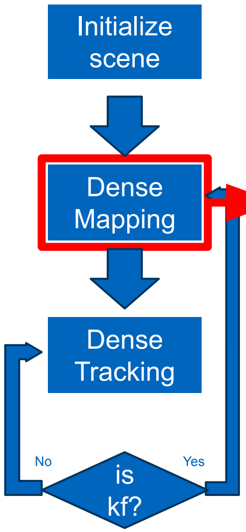
unproject pixel at frame r

transform to frame m

project into frame m



Method: Dense Mapping

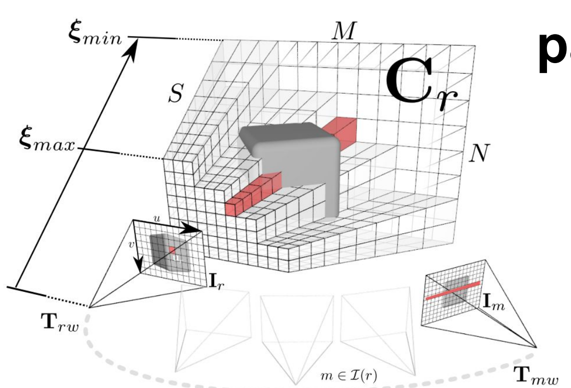


$$\rho_r(\mathbf{I}_m, \mathbf{u}, d) = \mathbf{I}_r(\mathbf{u}) - \mathbf{I}_m\left(\pi\left(\mathrm{KT}_{mr}\pi^{-1}(\mathbf{u}, d)\right)\right)$$

Hidden assumption: Brightness Constancy

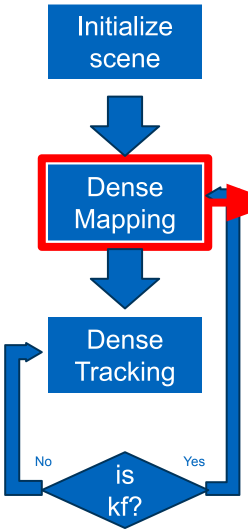
Problem:
photometric loss does not work in large baseline

patch-based intensity normalization?



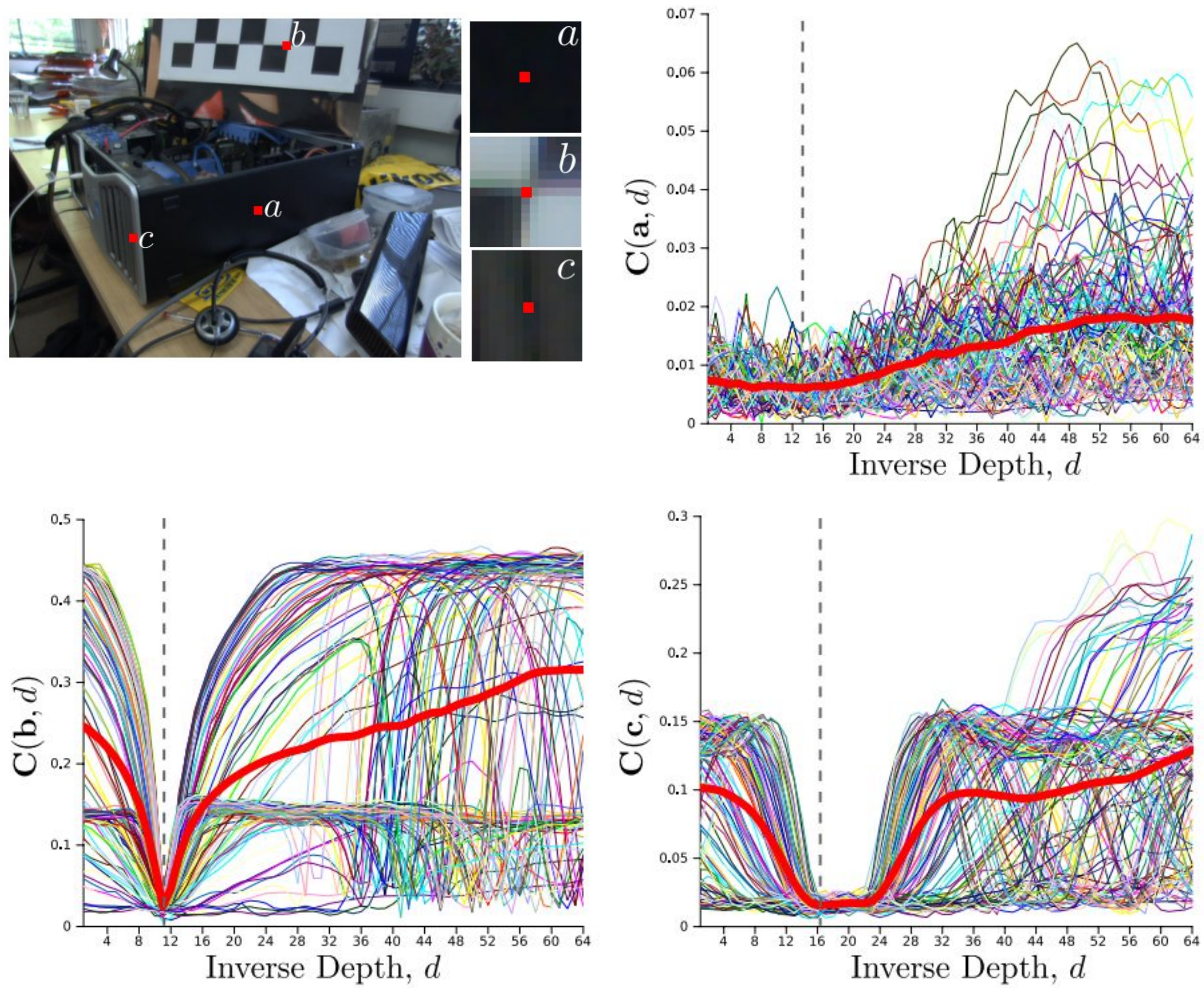
Solution:
manually keep baseline small

Method: Dense Mapping

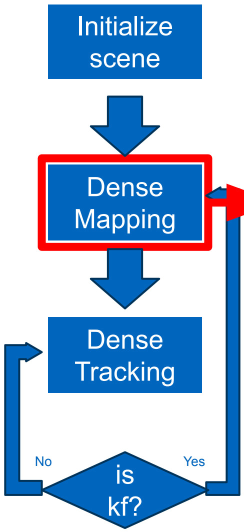


$$\mathbf{C}_r(\mathbf{u}, d) = \frac{1}{|\mathcal{I}(r)|} \sum_{m \in \mathcal{I}(r)} \|\rho_r(\mathbf{I}_m, \mathbf{u}, d)\|_1$$

Cost at different position

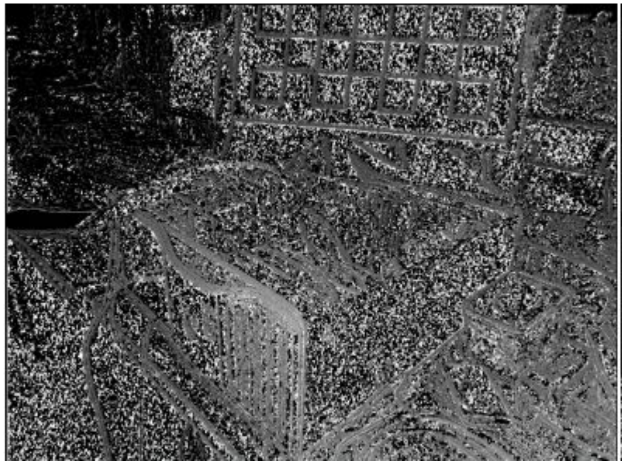


Method: Dense Mapping

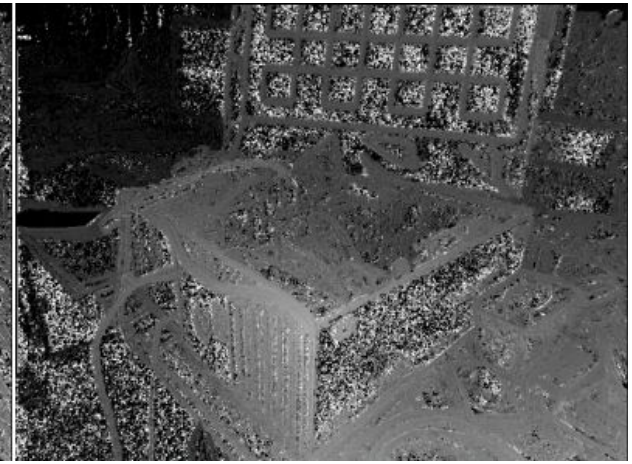


Depth map with different overlapping images number

$$\mathbf{C}_r(\mathbf{u}, d) = \frac{1}{|\mathcal{I}(r)|} \sum_{m \in \mathcal{I}(r)} \|\rho_r\left(\mathbf{I}_m, \mathbf{u}, d\right)\|_1$$

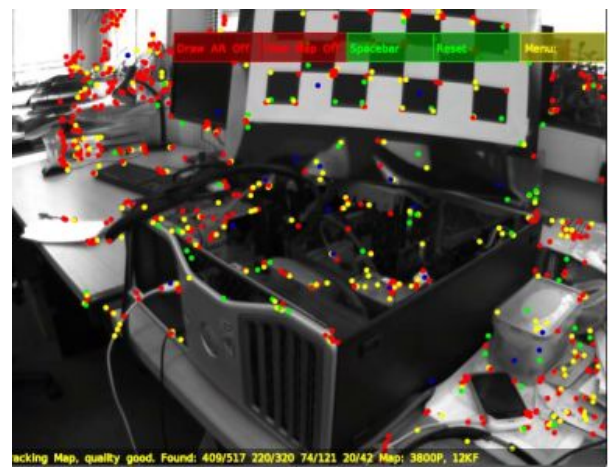


overlapping 2 images



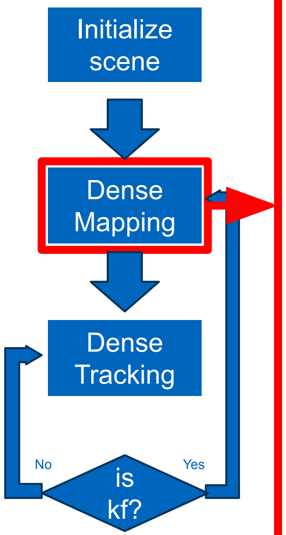
overlapping 10 images

overlapping 30 images



PTAM

Method: Dense Mapping



$$E_{\boldsymbol{\xi}} = \int_{\Omega} \left\{ g(\mathbf{u}) \| \nabla \boldsymbol{\xi}(\mathbf{u}) \|_{\epsilon} + \lambda \mathbf{C}\left(\mathbf{u}, \boldsymbol{\xi}(\mathbf{u})\right) \right\} d\mathbf{u}$$

regularizer

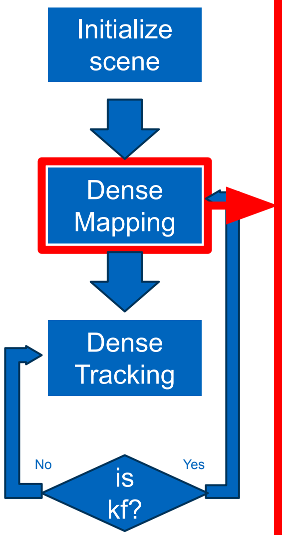
$$g(\mathbf{u}) = e^{-\alpha \| \nabla \mathbf{I}_r(\mathbf{u}) \|_2^{\beta}}$$

per pixel weight:
small weight at large intensity gradient
keep sharp edge

$\| \nabla \boldsymbol{\xi}(\mathbf{u}) \|_{\epsilon}$ total variation of inverse depth with Huber loss

$$\|x\|_{\epsilon} = \begin{cases} \frac{\|x\|_2^2}{2\epsilon} & \text{if } \|x\|_2 \leq \epsilon \\ \|x\|_1 - \frac{\epsilon}{2} & \text{otherwise} \end{cases}$$

Method: Dense Mapping



How to solve this non-convex loss?

$$E_{\boldsymbol{\xi}} = \int_{\Omega} \left\{ g(\mathbf{u}) \|\nabla \boldsymbol{\xi}(\mathbf{u})\|_{\epsilon} + \lambda \mathbf{C}\left(\mathbf{u}, \boldsymbol{\xi}(\mathbf{u})\right) \right\} \mathrm{d}\mathbf{u}$$

introduce an auxiliary variable $\boldsymbol{\alpha} : \Omega \to \mathbb{R}$

$$\mathbf{E}_{\boldsymbol{\xi},\boldsymbol{\alpha}} = \int_{\Omega} \left\{ g(\mathbf{u}) \|\nabla \boldsymbol{\xi}(\mathbf{u})\|_{\epsilon} + \frac{1}{2\theta} \left(\boldsymbol{\xi}(\mathbf{u}) - \boldsymbol{\alpha}(\mathbf{u}) \right)^2 + \lambda \mathbf{C}\left(\mathbf{u}, \boldsymbol{\alpha}(\mathbf{u})\right) \right\} \mathrm{d}\mathbf{u} \,.$$

coupling term

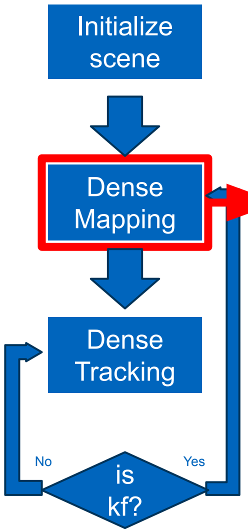
$$\mathbf{Q}(\mathbf{u}) = \frac{1}{2\theta} \left(\boldsymbol{\xi}(\mathbf{u}) - \boldsymbol{\alpha}(\mathbf{u}) \right)^2$$

Solution: Two Steps

1. first minimize $g(\mathbf{u}) \|\nabla \boldsymbol{\xi}(\mathbf{u})\|_{\epsilon} + \mathbf{Q}(\mathbf{u})$, find optimal $\boldsymbol{\xi}$
2. then perform exhaustive search of $\boldsymbol{\alpha}$ to minimize

$$\mathbf{Q}(\mathbf{u}) + \lambda \mathbf{C}\left(\mathbf{u}, \boldsymbol{\alpha}(\mathbf{u})\right)$$

Method: Dense Mapping

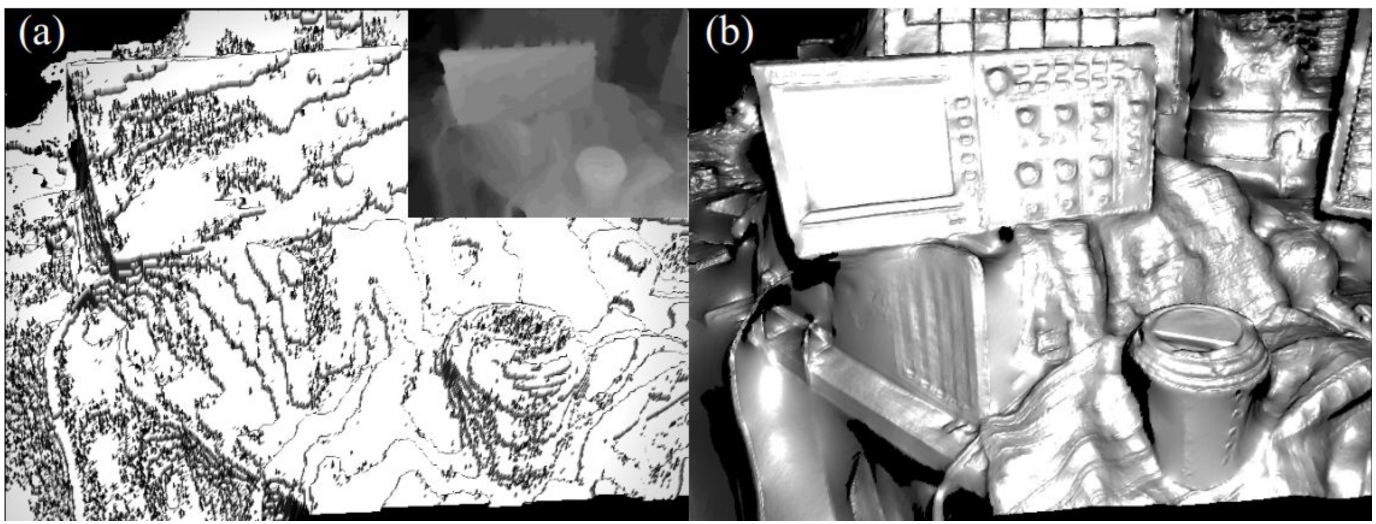


What else can we do to get more accurate depth?

We can embed a single Newton step in each iteration to achieve sub-sample refinement:

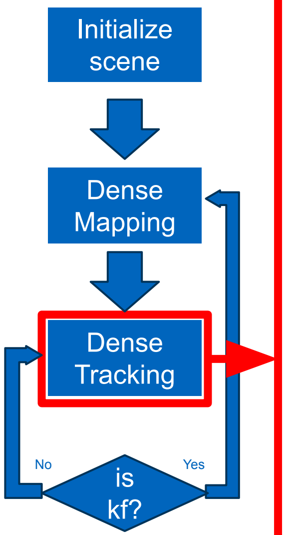
$$\hat{a}_{\mathbf{u}}^{n+1} = a_{\mathbf{u}}^{n+1} - \frac{\nabla \mathbf{E}^{\text{aux}}(\mathbf{u}, d_{\mathbf{u}}^{n+1}, a_{\mathbf{u}}^{n+1})}{\nabla^2 \mathbf{E}^{\text{aux}}(\mathbf{u}, d_{\mathbf{u}}^{n+1}, a_{\mathbf{u}}^{n+1})}$$

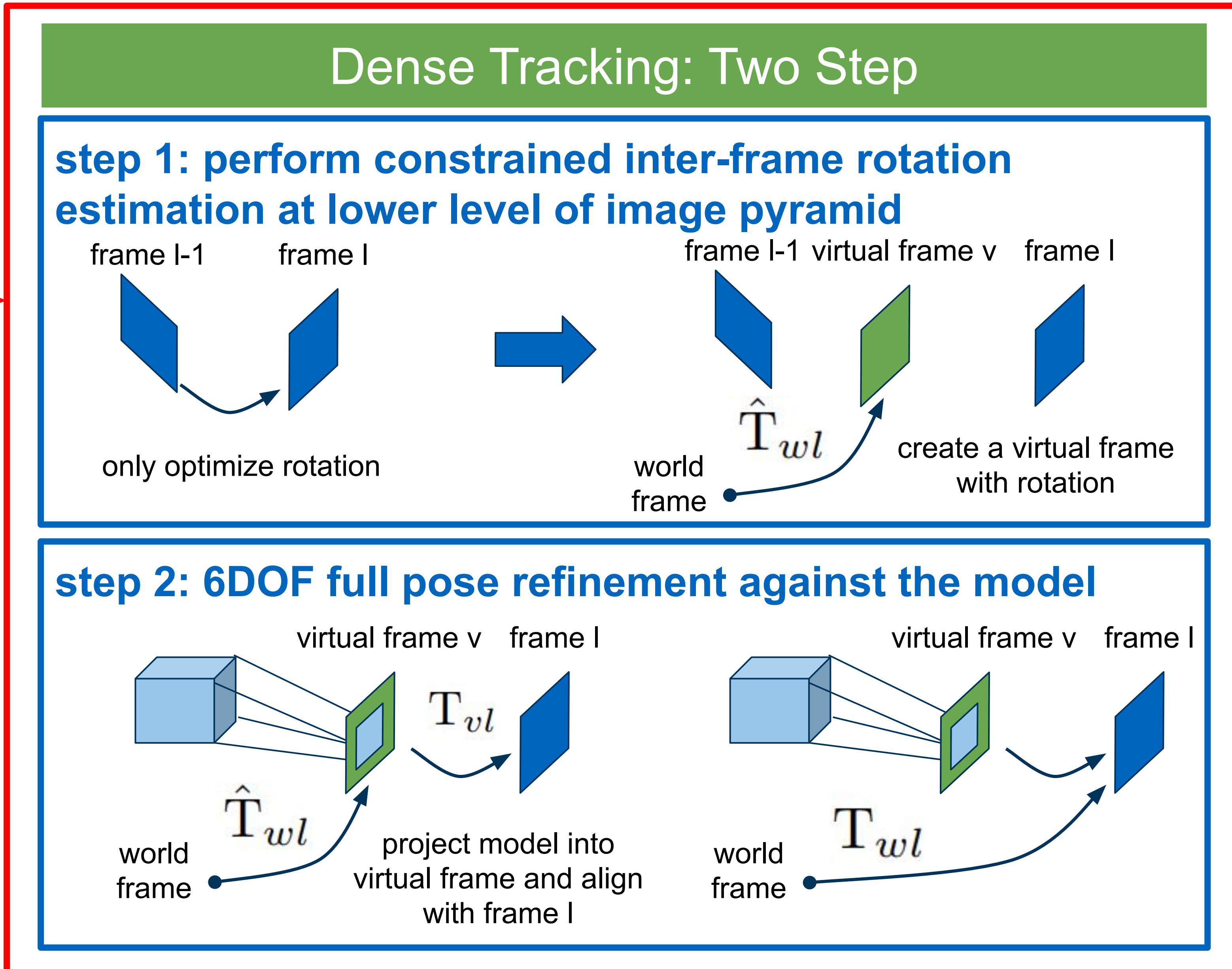
$$\mathbf{E}^{\text{aux}}(\mathbf{u}, d_{\mathbf{u}}, a_{\mathbf{u}}) = \frac{1}{2\theta}(d_{\mathbf{u}} - a_{\mathbf{u}})^2 + \lambda \mathbf{C}(\mathbf{u}, a_{\mathbf{u}})$$



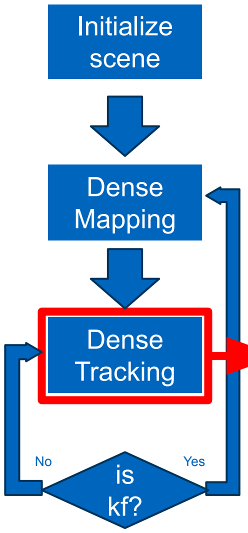
without sub-sample refinement | with sub-sample refinement

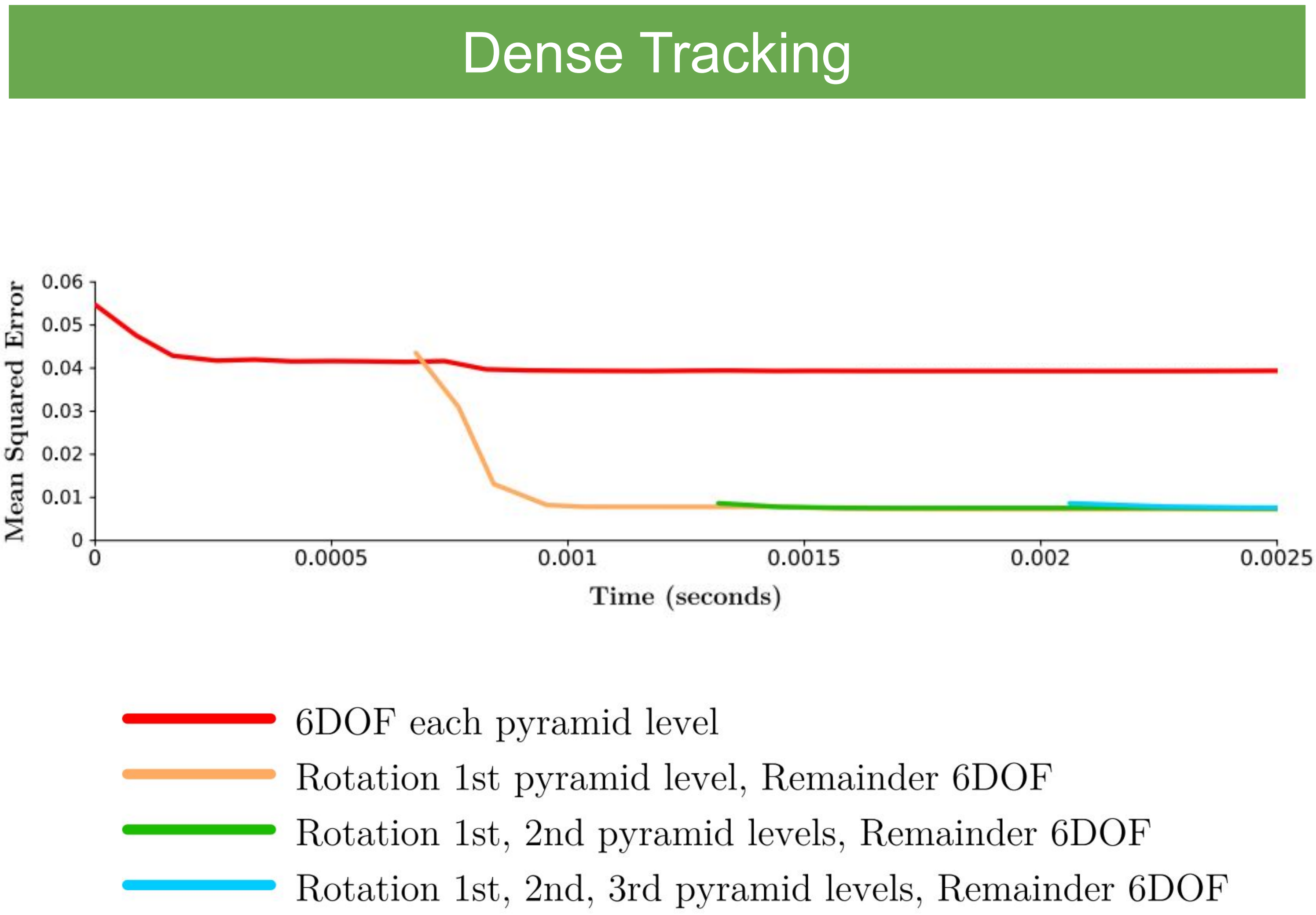
Method: Dense Tracking





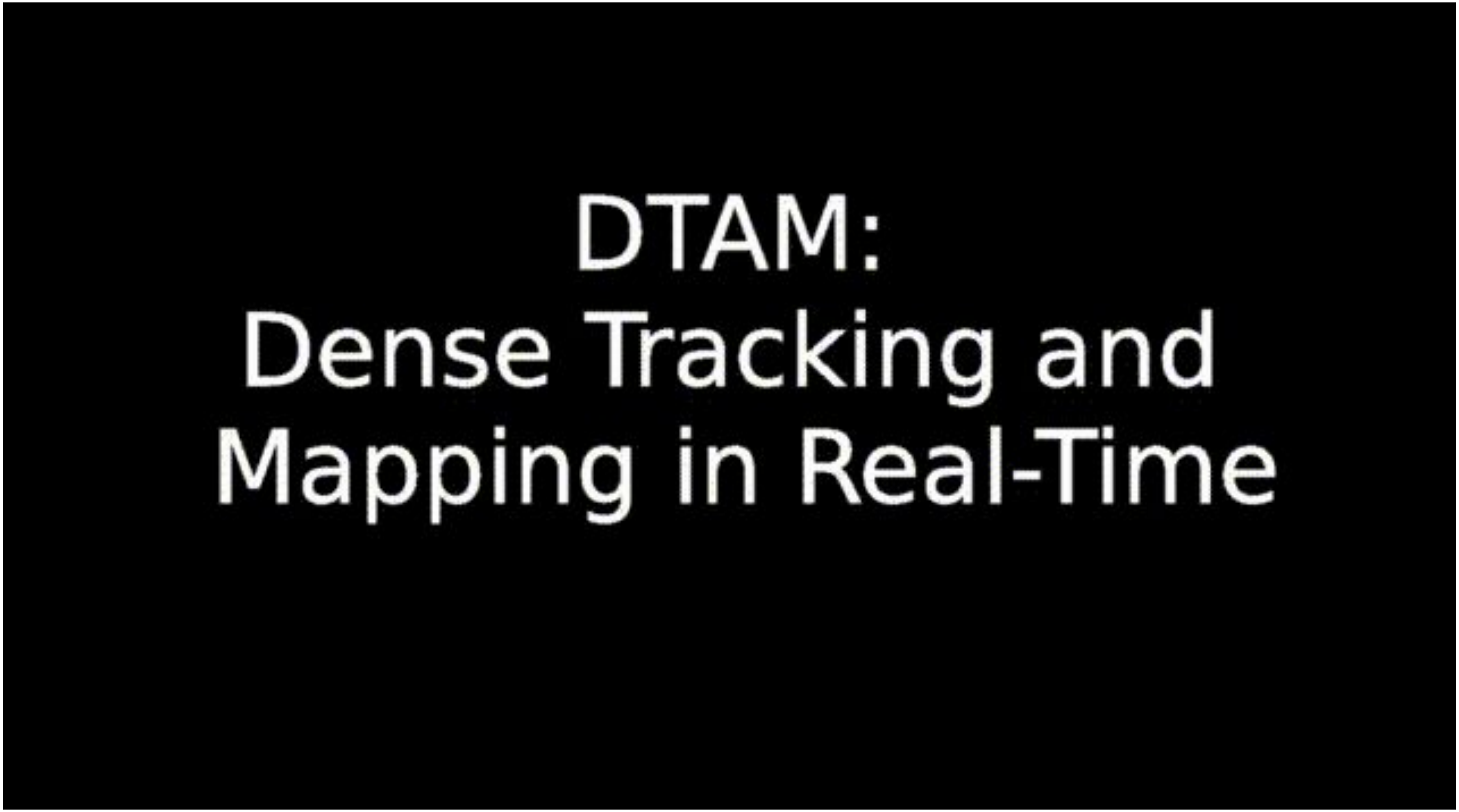
Method: Dense Tracking





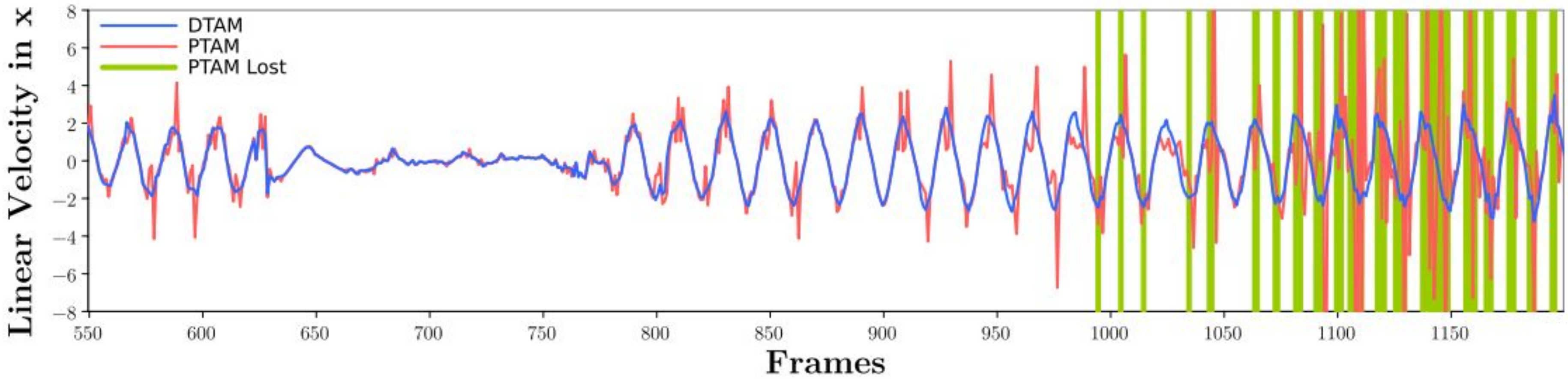
Evaluation

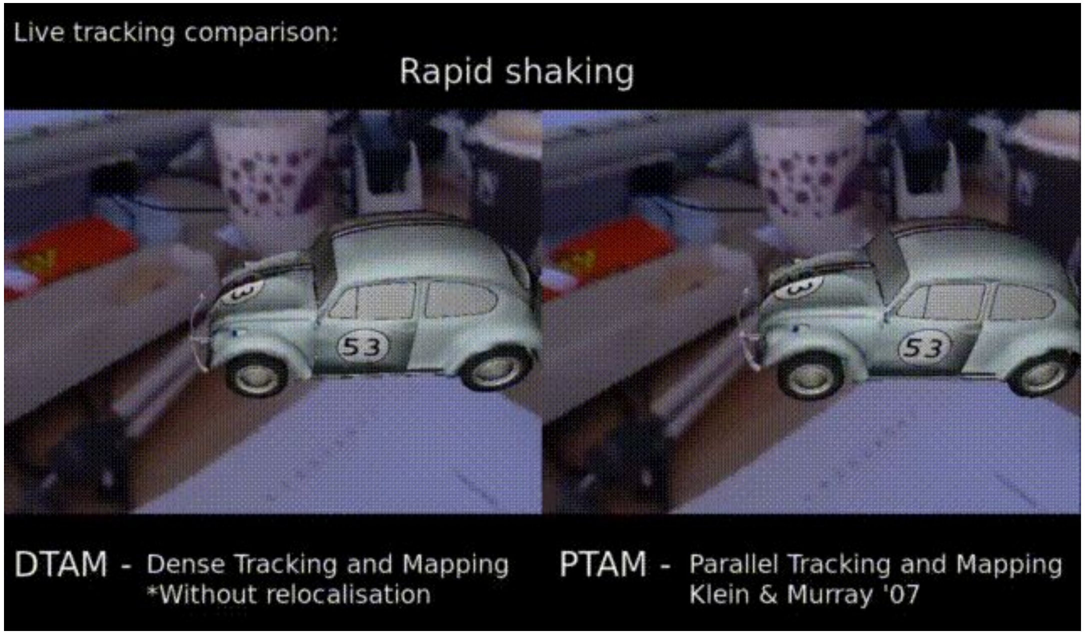
How this method works?



Evaluation

Robustness under vibration





Evaluation

Robustness under defocus

Live tracking comparison:

DTAM - Dense Tracking and Mapping
*Without relocalisation

PTAM - Parallel Tracking and Mapping
Klein & Murray '07

Evaluation

Robustness under rapid translation

Thank you for your attention