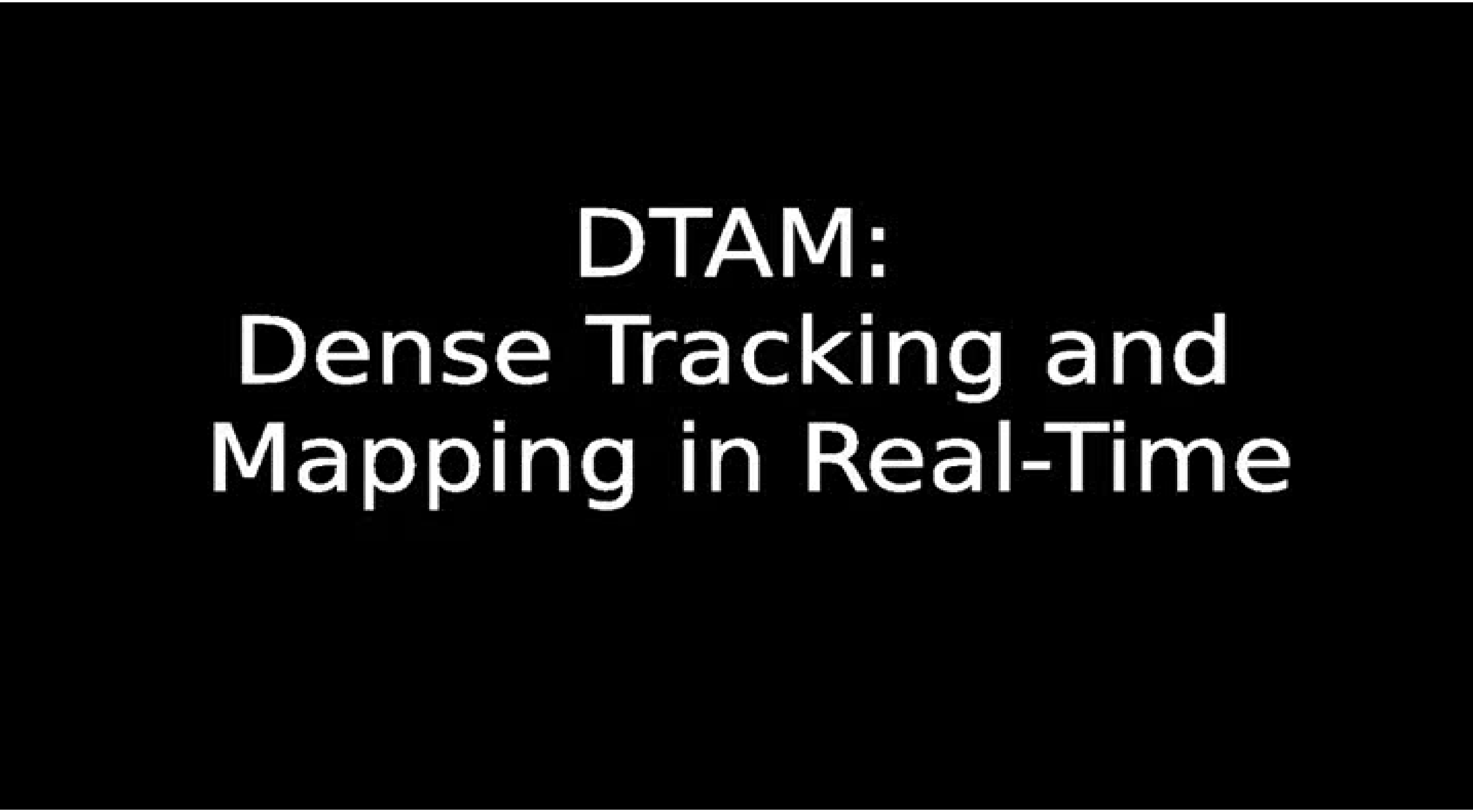
University of Zurich[UZH] | ETH zürich
Institute of Informatics – Institute of Neuroinformatics
ROBOTICS & PERCEPTION GROUP

Lecture 12a
Dense 3D Reconstruction

Davide Scaramuzza
http://rpg.ifi.uzh.ch/

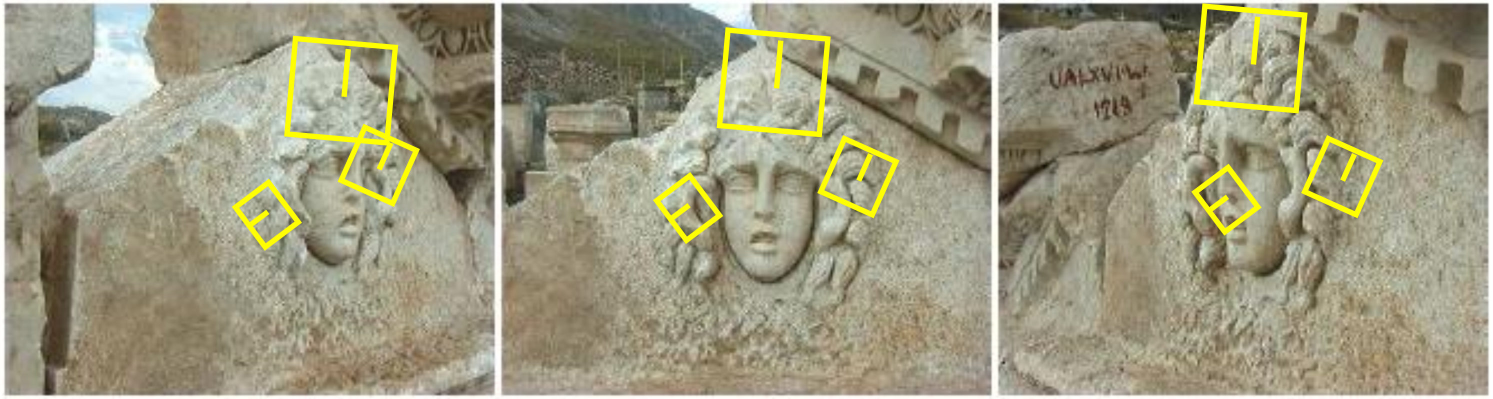
DTAM: Dense Tracking and Mapping in Real-Time, ICCV’11 by Newcombe, Lovegrove, Davison

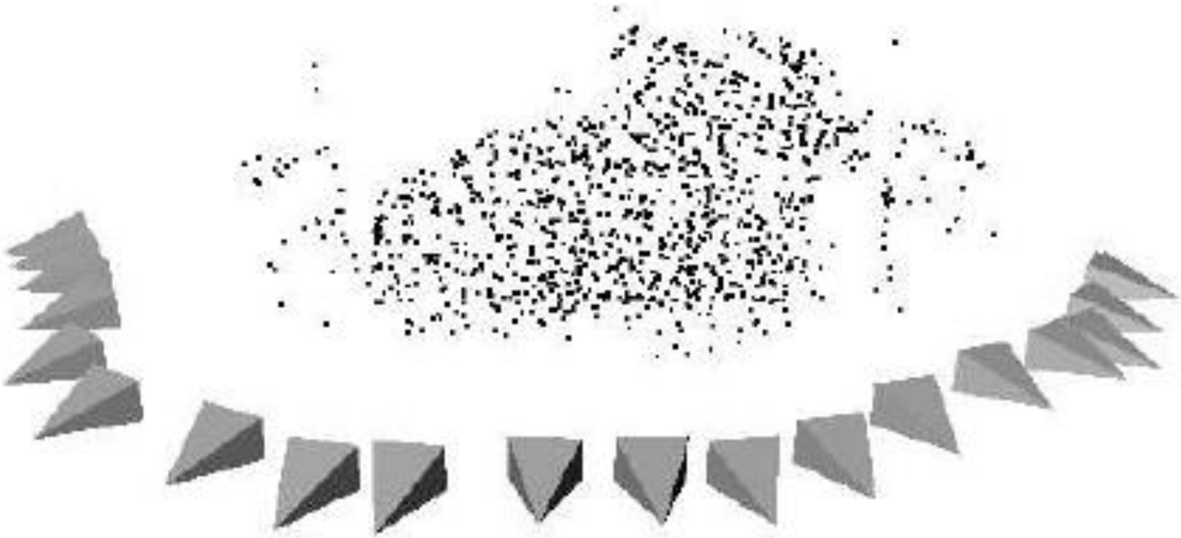


https://youtu.be/Df9WhgibCQA

Sparse Reconstruction

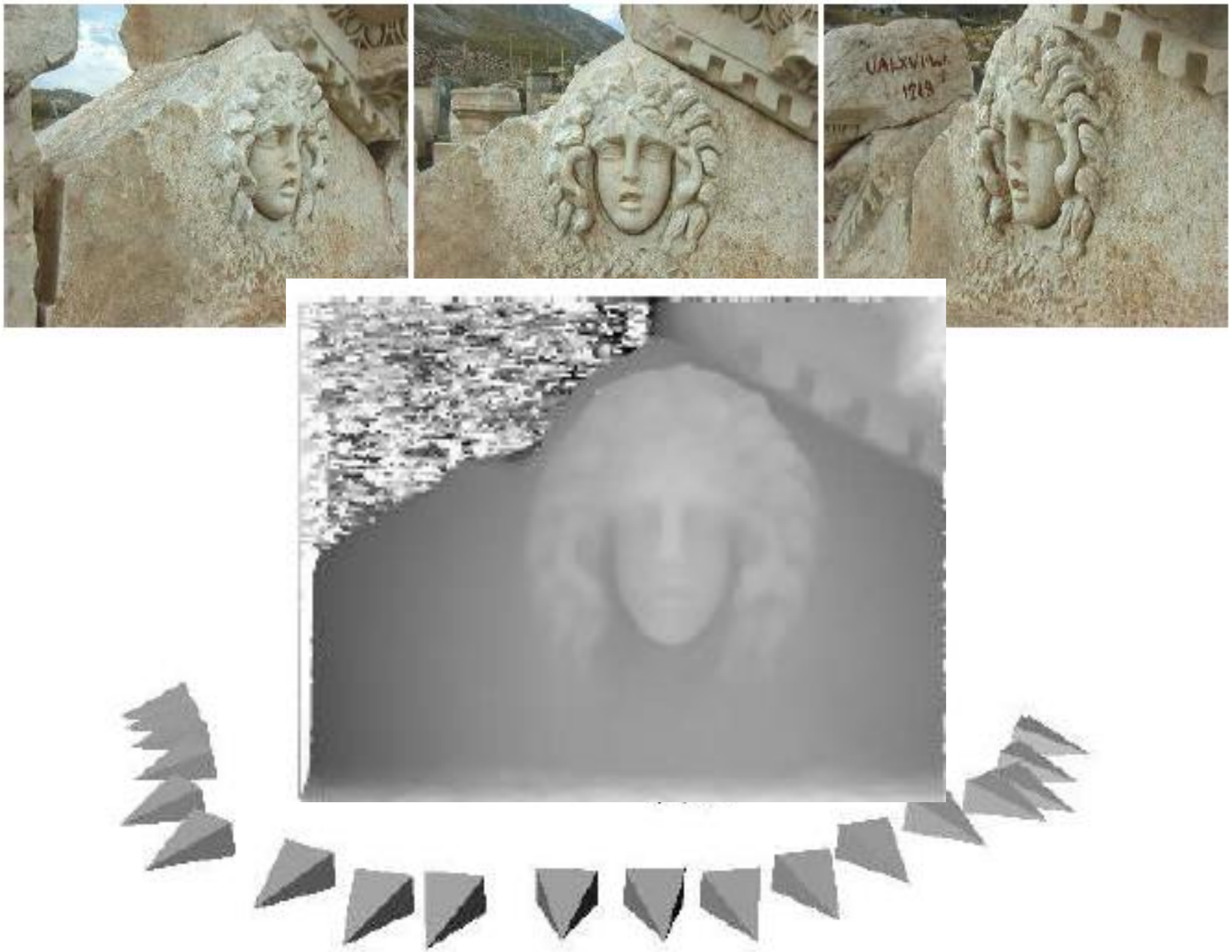
- Estimate the structure from a “sparse” set of features





Dense Reconstruction

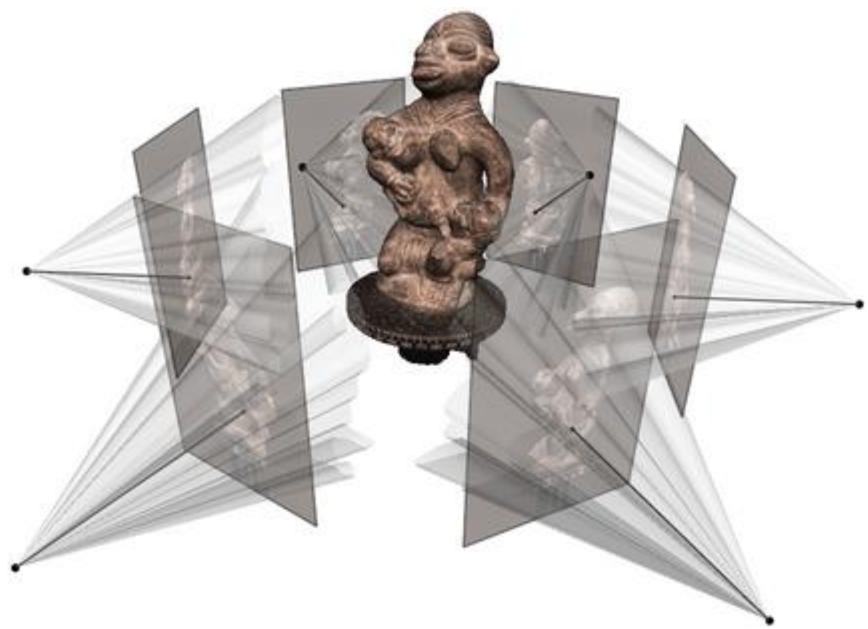
- Estimate the structure from a “dense” region of pixels



Dense Reconstruction (or Multi-view stereo)

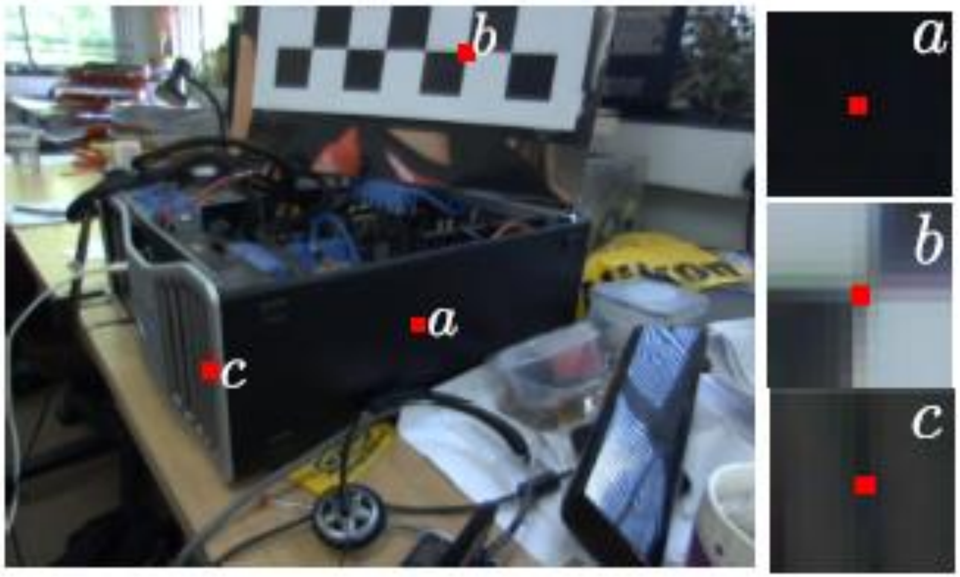
Problem definition:

- **Input**: calibrated images from several viewpoints (i.e., K, R, T are known for each camera, e.g., from SFM)
- **Output**: 3D object **dense** reconstruction



Challenges

- Dense reconstruction requires establishing dense correspondences
- But not all pixels can be matched reliably: Flat regions, edges, viewpoint and illumination changes, occlusions



[Newcombe et al. 2011]

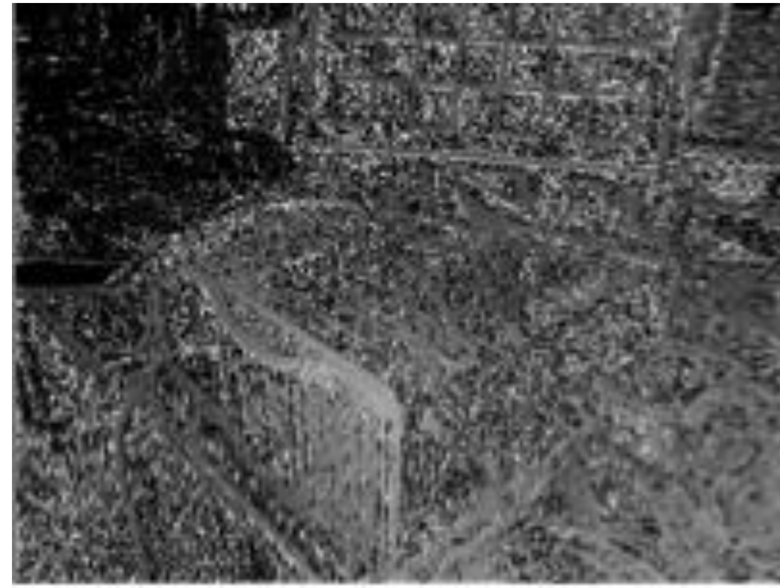
Idea: Take advantage of many small-baseline views where high quality matching is possible

Dense reconstruction workflow

Step 1: Local methods

- Estimate depth independently for each pixel

how do we compute correspondences for *every* pixel?

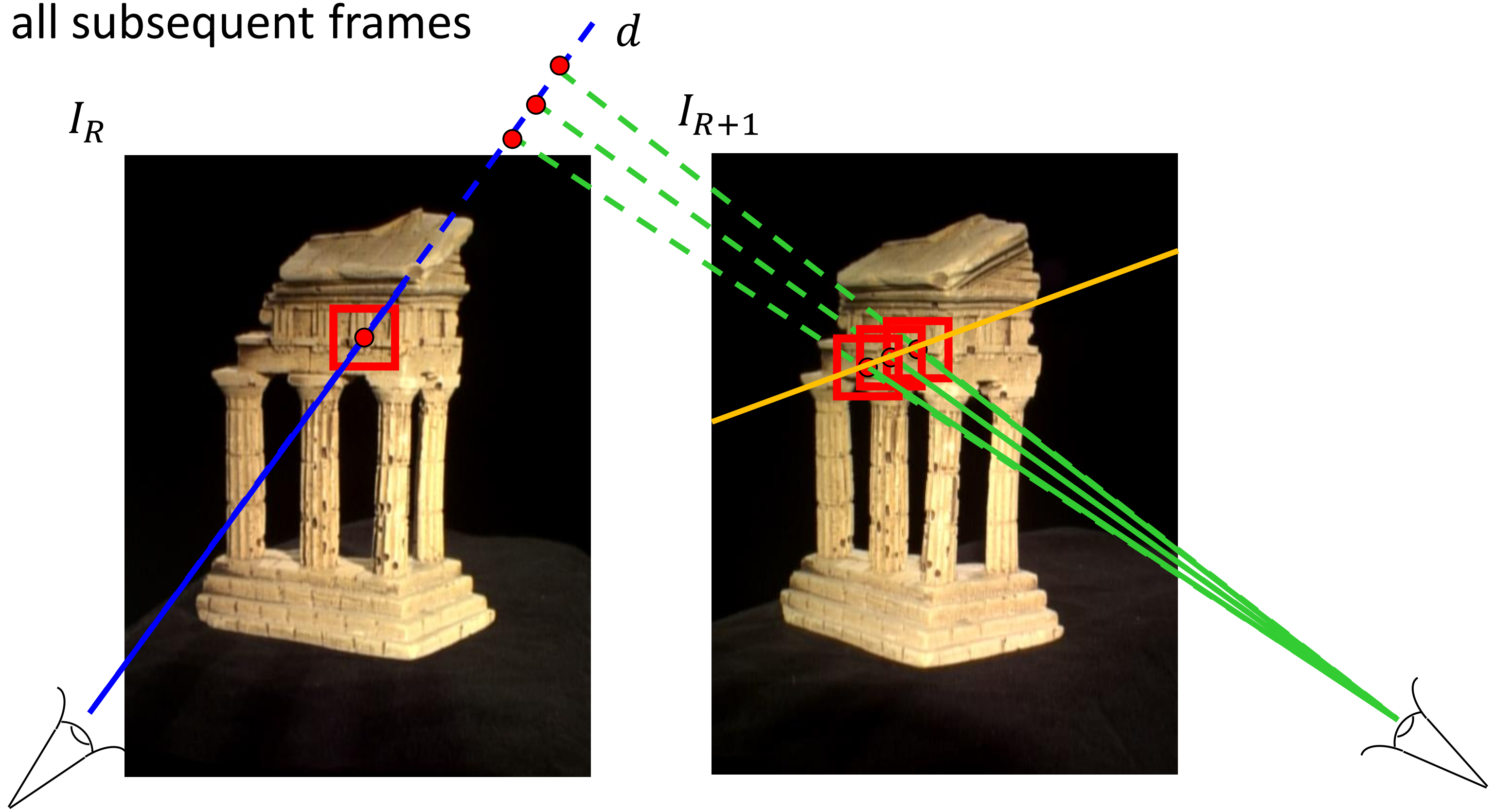


Step 2: Global methods

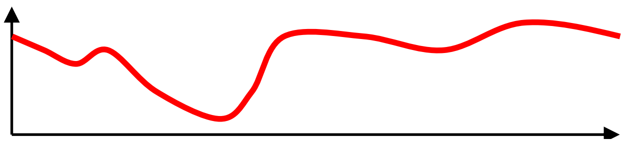
- Refine the *depth map as a* whole by enforcing smoothness. This process is called *regularization*

Solution: Aggregated Photometric Error

Set the first image as reference and estimate depth at each pixel by minimizing the Aggregated Photometric Error in all subsequent frames



Photometric error: $\rho\left(I_R(u,v) - I_{R+1}(u',v',d)\right)$

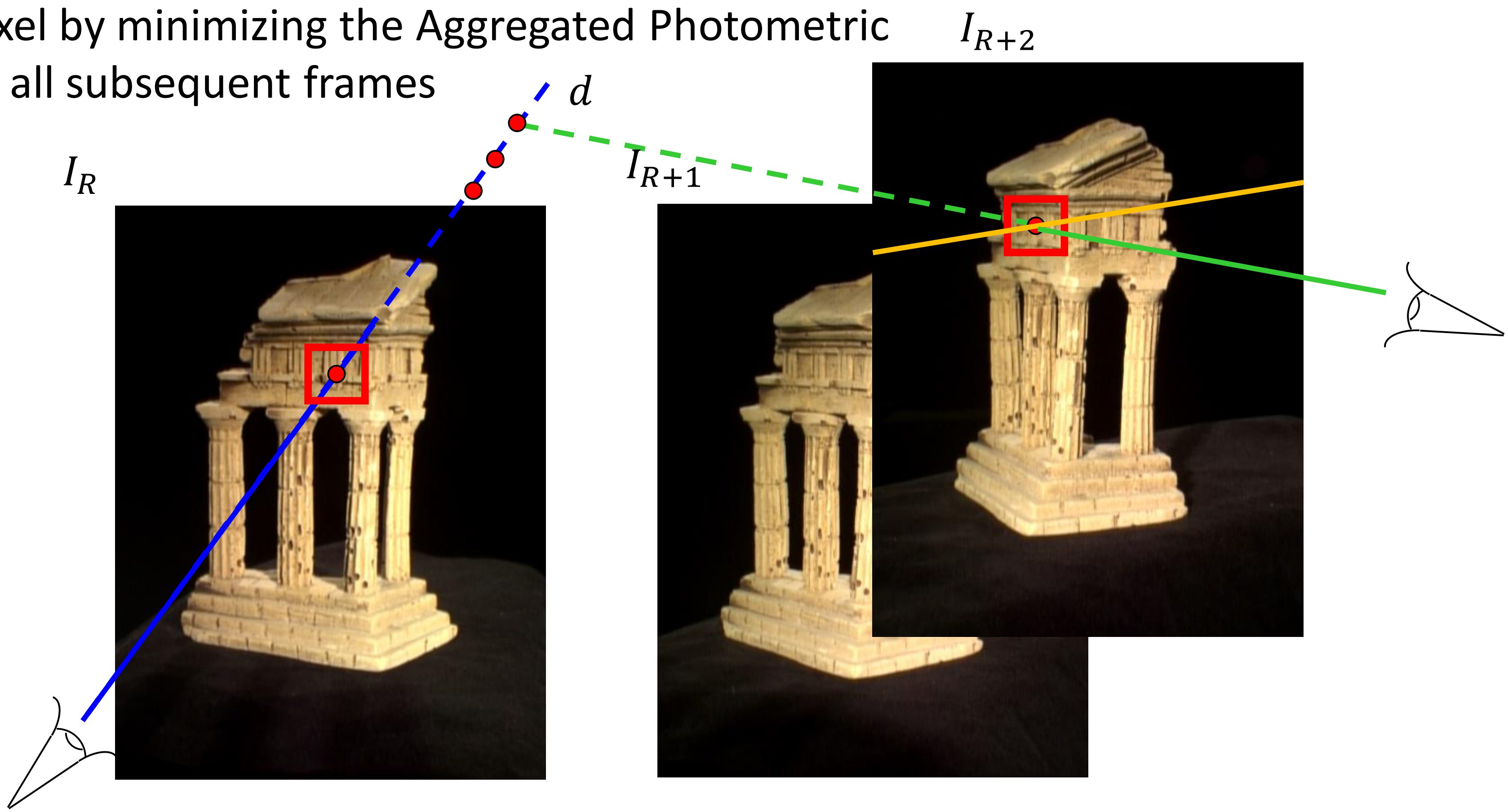


Depth of pixel (u,v) in I_R

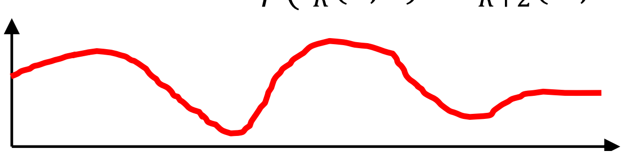
This error term is computed for between the reference image and each subsequent frame. The sum of these error terms is called Aggregated Photometric Error (see next slide)

Solution: Aggregated Photometric Error

Set the first image as reference and estimate depth at each pixel by minimizing the Aggregated Photometric Error in all subsequent frames



Photometric error: $\rho\big(I_R(u,v) - I_{R+2}(u',v',d)\big)$

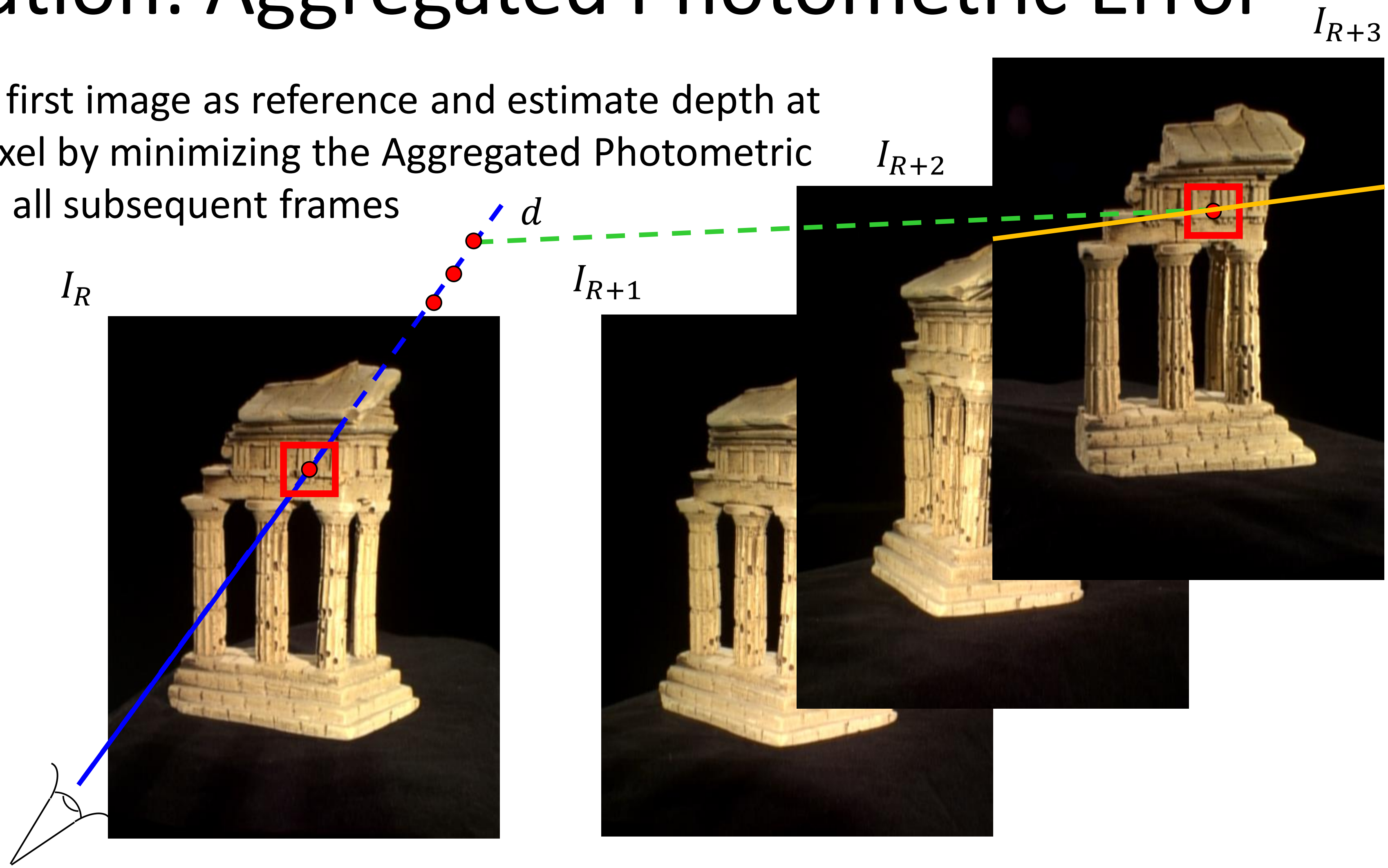


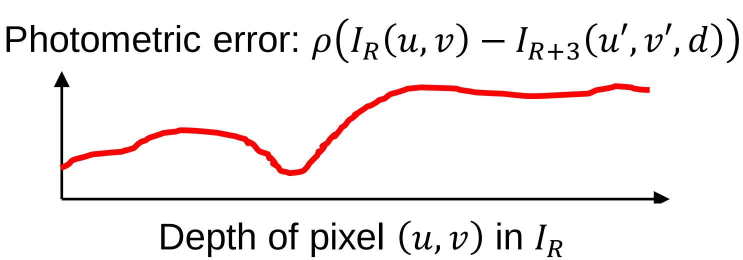
Depth of pixel (u,v) in I_R

This error term is computed for between the reference image and each subsequent frame. The sum of these error terms is called Aggregated Photometric Error (see next slide)

Solution: Aggregated Photometric Error

Set the first image as reference and estimate depth at each pixel by minimizing the Aggregated Photometric Error in all subsequent frames

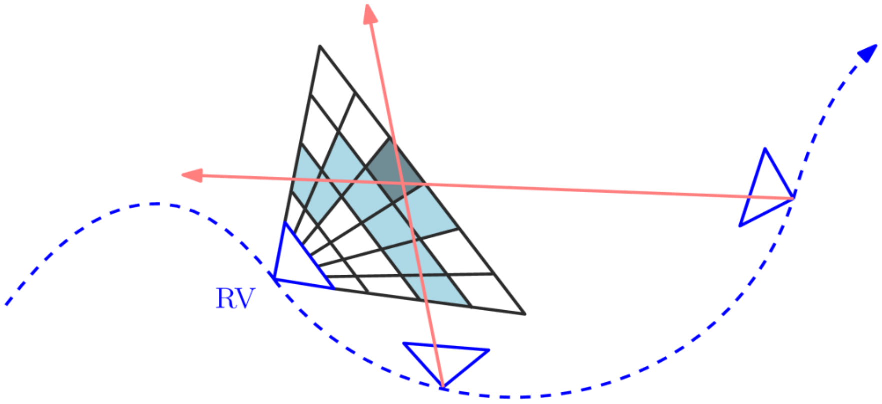




This error term is computed for between the reference image and each subsequent frame. The sum of these error terms is called Aggregated Photometric Error (see next slide)

Disparity Space Image (DSI)

Reference image



Non-uniform, projective grid,
centered on the reference frame I_R

- Image resolution: 240x180 pixels
- Number of disparity (depth) levels: 100
- DSI:
  - size: 240x180x100 voxels;
    each voxel contains the Aggregated Photometric Error $C(u, v, d)$ (see next slide)
  - white = high Aggregated Photometric Error
  - blue = low Aggregated Photometric Error

Disparity Space Image (DSI)

Reference image

DSI (dark means high)

240 x 180 x 100 voxels

Disparity Space Image (DSI)

- For a given image point (u, v) and for discrete depth hypotheses d, the **Aggregated Photometric Error** $C(u, v, d)$ with respect to the reference image I_R can be stored in a volumetric 3D grid called the **Disparity Space Image (DSI)**, where each voxel has value:

$$C(u, v, d) = \sum_{k=R+1}^{R+n-1} \rho\big(I_R(u, v) - I_k(u', v', d)\big)$$

Where n is the number of images considered and $I_k(u', v', d)$ is the patch of intensity values in the k-th image centered on the pixel (u', v') corresponding to the patch $I_R(u, v)$ in the reference image I_R and depth hypothesis d; thus, formally:

$$I_k(u', v', d) = I_k\left(\pi\left(T_{k,R}(\pi^{-1}(u, v) \cdot d)\right)\right)$$

where $T_{k,R}$ is the relative pose between frames R and K

- $\rho(\cdot)$ is the photometric error (SSD) (e.g. L_1, L_2, Tukey, or Huber norm)

Depth estimation

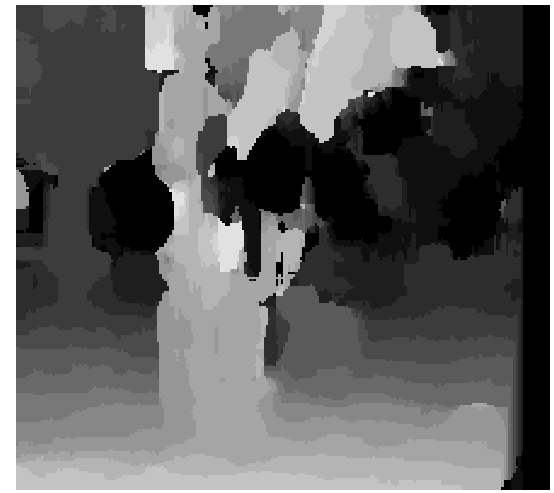
The solution to the depth estimation problem is to find a **function $\boldsymbol{d(u,v)}$** (called ***"depth map"***) in the DSI that satisfies minimizes the **aggregated photometric error:**

$$d(u,v) = \ arg \min_{d} \sum_{(u,v)} C(u,v,d(u,v))$$

Effects of the patch **size** on the resulting depth map

The computation of the aggregated photometric error depends on the patch size

$W = 3$



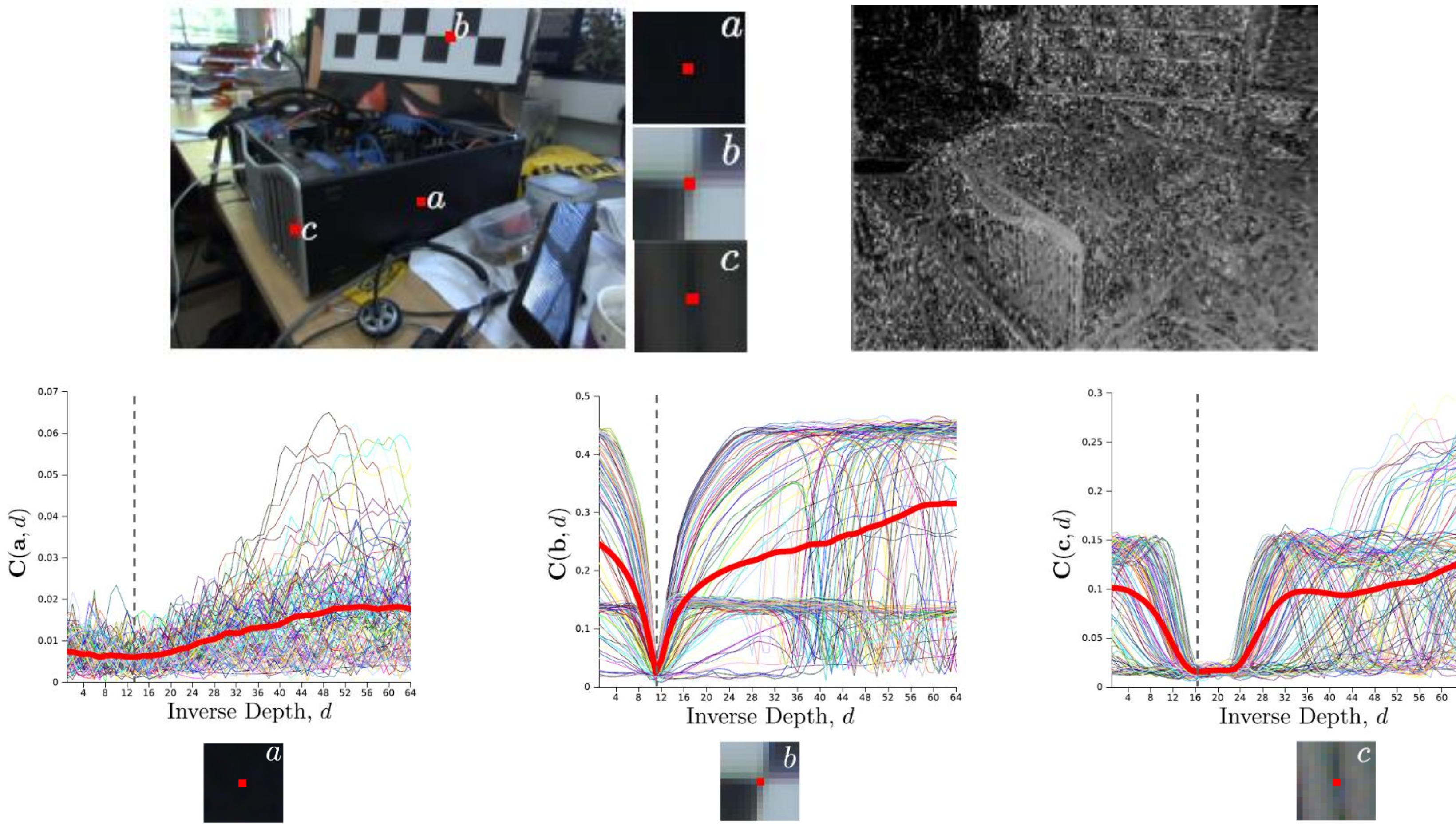
$W = 20$

- Smaller window
  - \+ More detail
  - – More noise

Can we use a patch size of 1×1 pixels?

- Larger window
  - \+ Smoother disparity maps
  - – Less detail

Effects of the patch **appearance** on the resulting depth map



- Aggregated photometric error for flat regions (a) and *edges parallel to the epipolar line* (c) show flat valleys (plus noise)
- For distinctive features (corners as in (b) or blobs), the aggregated photometric error has one clear minimum.
- Non distinctive features (e.g., from repetitive texture) will show multiple minima

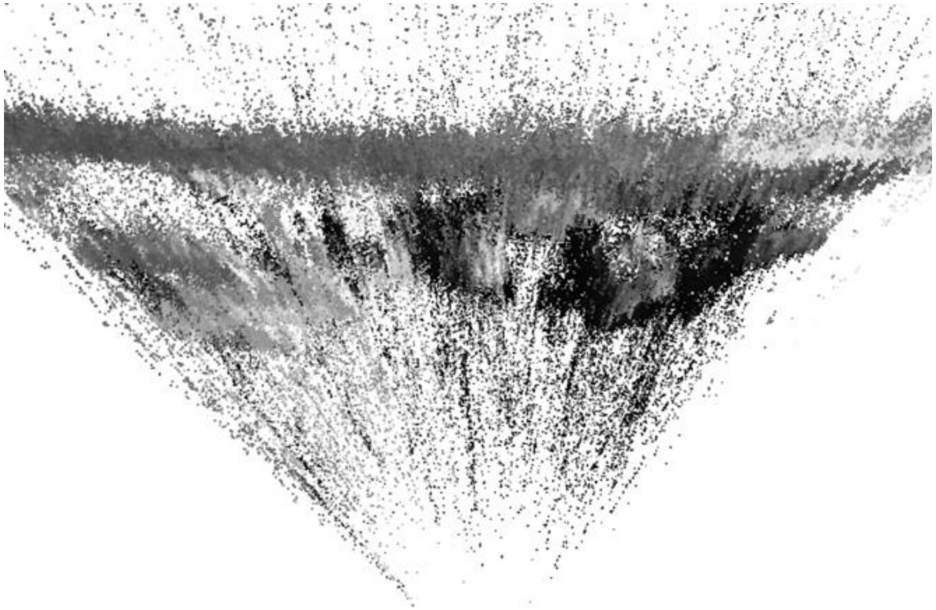
Regularization

To penalize non smooth reconstructions, due to image noise and ambiguous texture, we add a smoothing term (called regularization) to the optimization:

$$d(u,v) = \ arg \min_{d} \textstyle\sum_{(u,v)} C(u,v,d(u,v)) \quad \text{(local methods)}$$

subject to

Piecewise smooth (global methods)



First reconstruction via local methods

Regularization

To penalize non smooth reconstructions, due to image noise and ambiguous texture, we add a smoothing term (called regularization) to the optimization:

$$d(u,v) = \arg\min_{d} \sum_{(u,v)} C(u,v,d(u,v)) \quad \text{(local methods)}$$

subject to

Piecewise smooth (global methods)

Effect of global methods: smoothing

Regularization

- Formulated in terms of energy minimization
- The objective is to find a *surface* $d(u,v)$ that minimizes a global energy

$$E(d) = \underbrace{E_d(d)}_{\text{Data term}} + \lambda \underbrace{E_s(d)}_{\text{Regularization term (i.e., smoothing)}}$$

Data term: $$E_d(d) = \sum_{(u,v)} C(u,v,d(u,v))$$

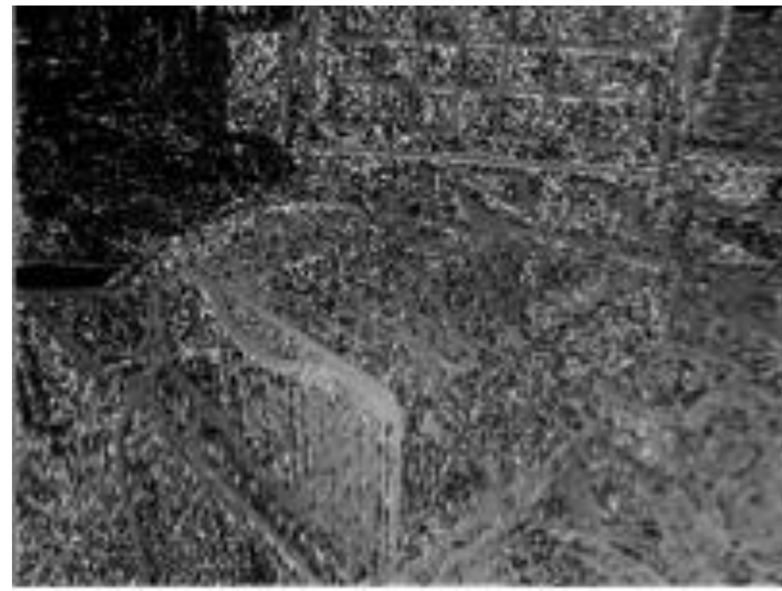
Regularization term: $$E_s(d) = \sum_{(u,v)} \left(\frac{\partial d)}{\partial u}\right)^2 + \left(\frac{\partial d)}{\partial v}\right)^2$$

where:

– λ controls the tradeoff data / regularization. What happens as λ increases?

Regularized depth maps

- **The regularization term** $E_s(d)$
  - ***Smooths*** *non smooth surfaces* (results of noisy measurements and ambiguous texture) as well as discontinuities
  - ***Fills the holes***



Final depth image for increasing λ
[Newcombe et al. 2011]

Regularized depth maps

- **The regularization term** $E_s(d)$
  - ***Smooths*** *non smooth surfaces* (results of noisy measurements and ambiguous texture) as well as discontinuities
  - ***Fills the holes***

Final depth image for increasing λ
[Newcombe et al. 2011]

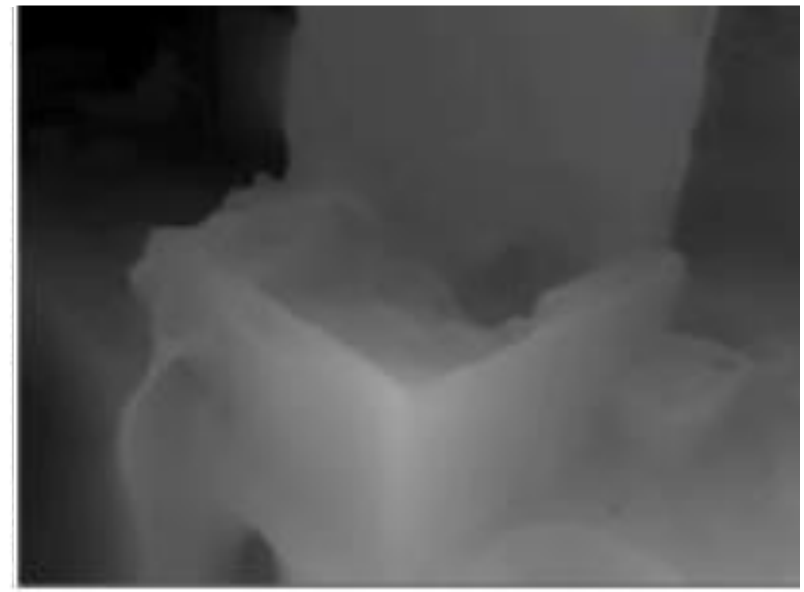
Regularized depth maps

- **The regularization term** $E_s(d)$
  - ***Smooths*** *non smooth surfaces* (results of noisy measurements and ambiguous texture) as well as discontinuities
  - ***Fills the holes***

Final depth image for increasing λ
[Newcombe et al. 2011]

Regularized depth maps

- **The regularization term** $E_s(d)$
  - ***Smooths*** *non smooth surfaces* (results of noisy measurements and ambiguous texture) as well as discontinuities
  - ***Fills the holes***



Final depth image for increasing λ
[Newcombe et al. 2011]

How to deal with actual scene depth discontinuities?

- **Problem:** since we don't know a priori where depth discontinuities are, we can make the following assumption:

depth *discontinuities* ***coincide*** with **intensity** *discontinuities* (i.e., image gradients)

- **Solution: control regularization term** according to image gradient

$$E_s(d) = \sum_{(u,v)} \left(\frac{\partial d)}{\partial u}\right)^2 \rho_I \left(\frac{\partial I)}{\partial u}\right)^2 + \left(\frac{\partial d)}{\partial v}\right)^2 \rho_I \left(\frac{\partial I)}{\partial v}\right)^2$$

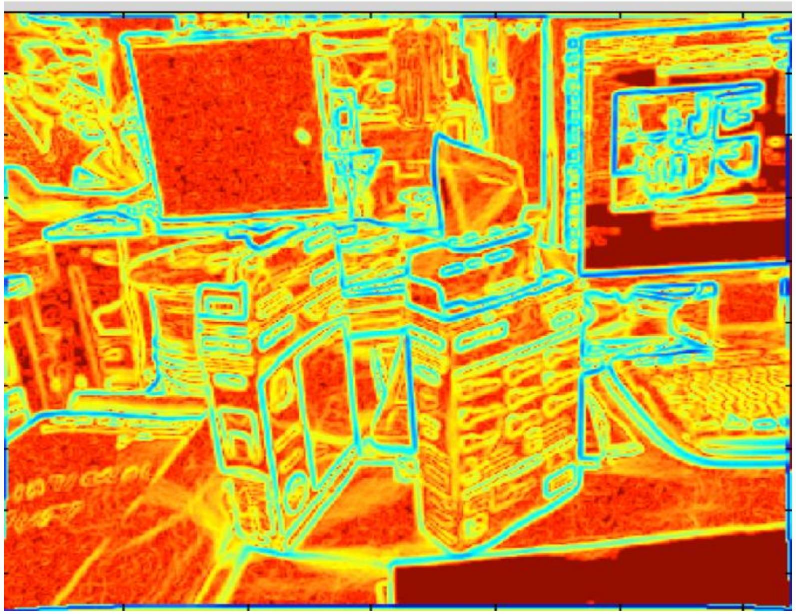
where ρ_I is a monotonically decreasing function (e.g., logistic) of image gradients:

- **high for small image gradients** (i.e., regularization term dominates)
- **low** for **high image gradients** (i.e., data term dominates)

Effect of ρ_I on intensity discontinuities

Reference image

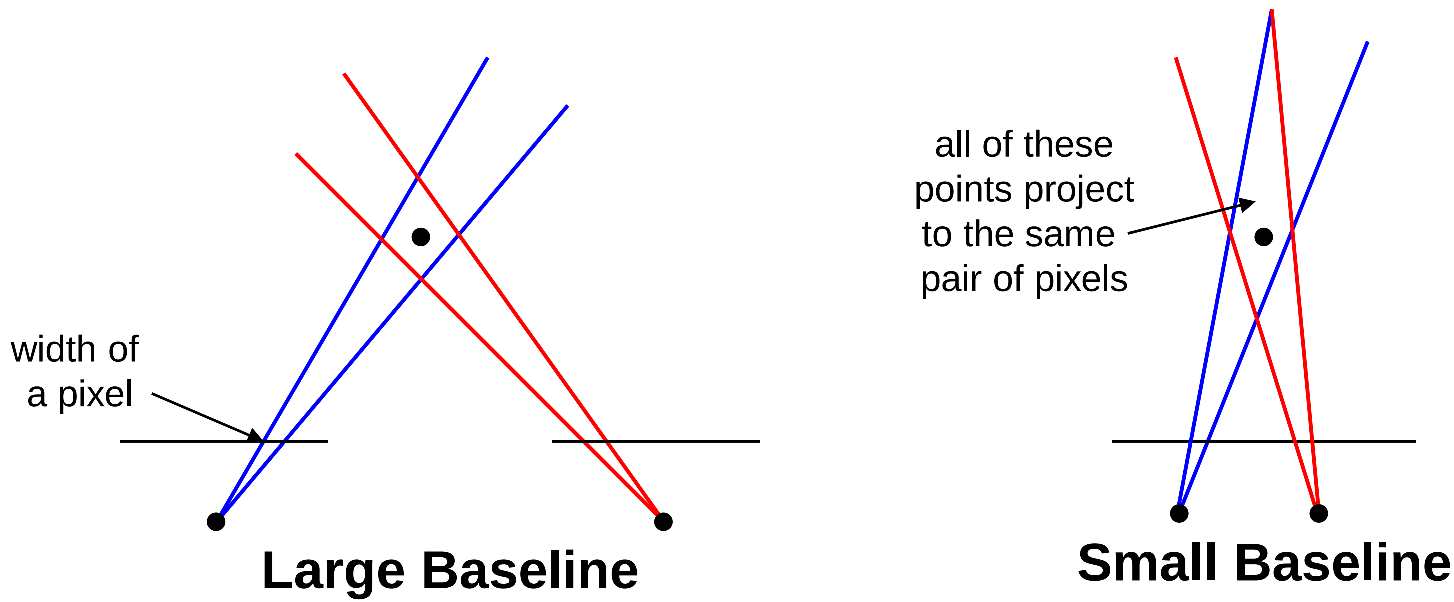
ρ_I (red means high)



where ρ_I is a monotonically decreasing function (e.g., logistic) of image gradients:

- **high for small image gradients** (i.e., regularization term dominates)
- **low** for **high image gradients** (i.e., data term dominates)

Choosing the baseline between subsequent frames



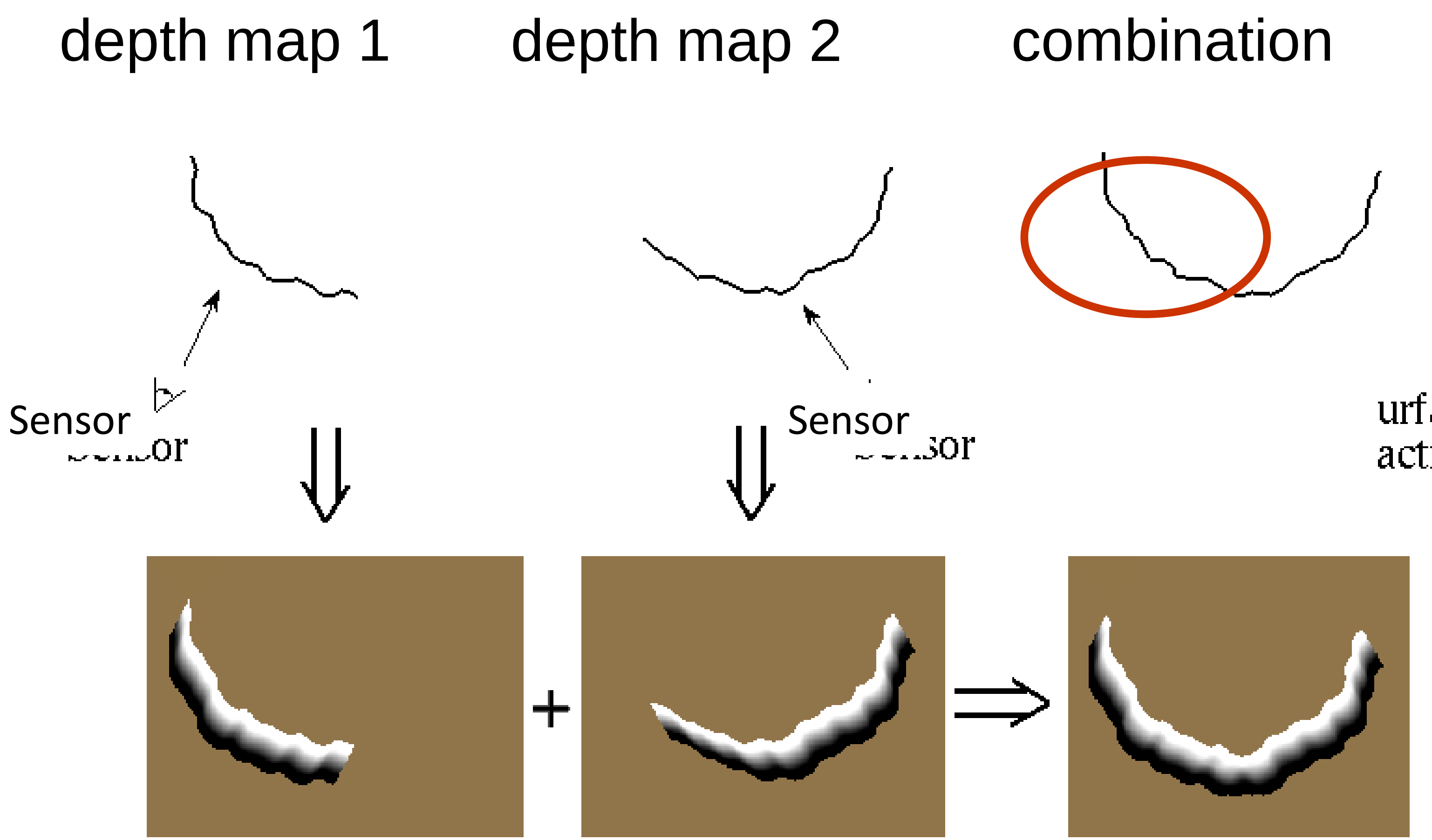
What's the optimal baseline ?

- Too large: ***difficult search problem*** due to wide view point changes
- Too small: ***large depth error***

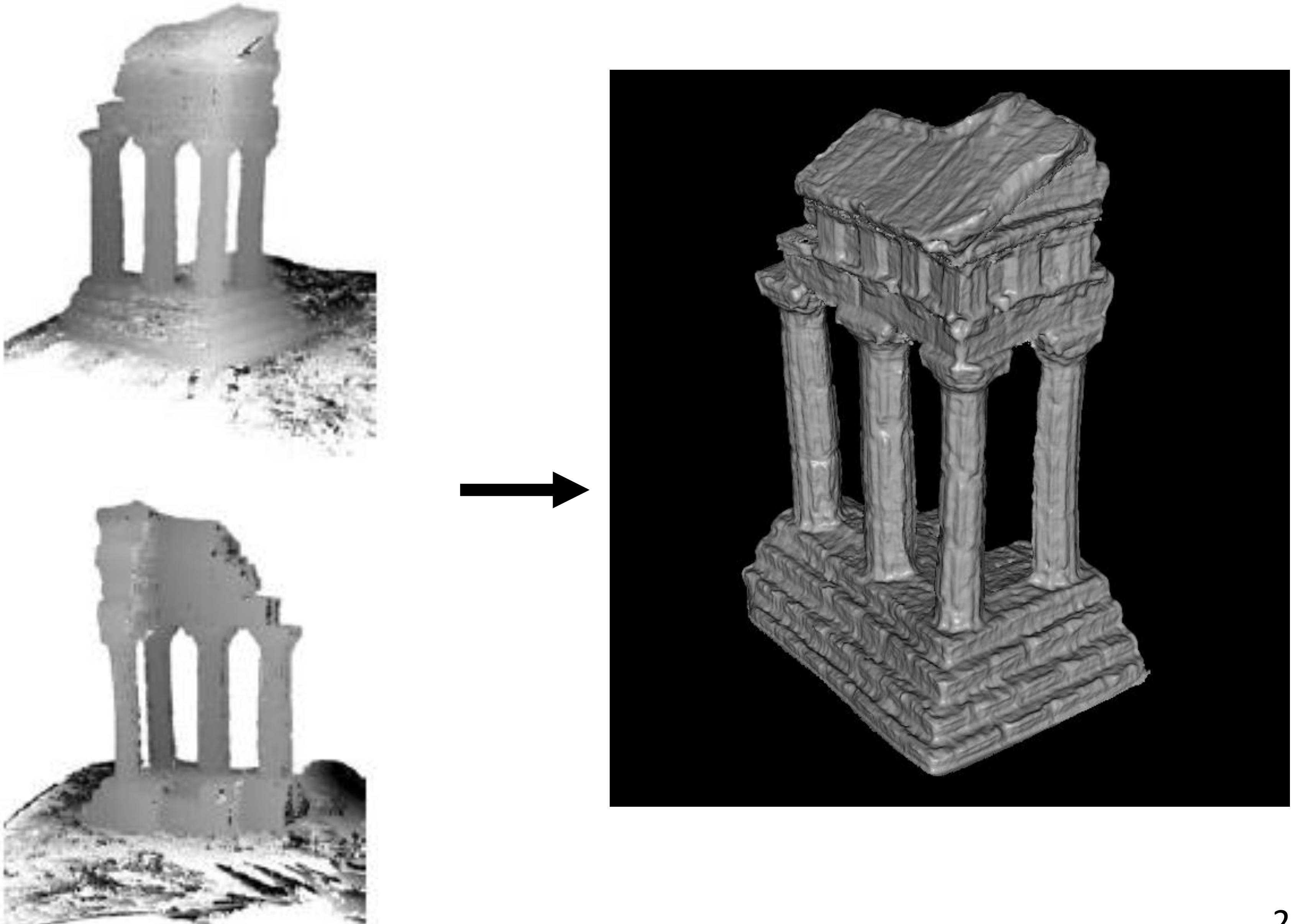
Solution

- Obtain depth map from **small baselines**
- When baseline becomes large (e.g., >10% of the avg scene depth), then **create new reference frame** (keyframe) and start a new depth map computation

Fusion of multiple depth maps



Fusion of multiple depth maps



Depth map fusion

input image

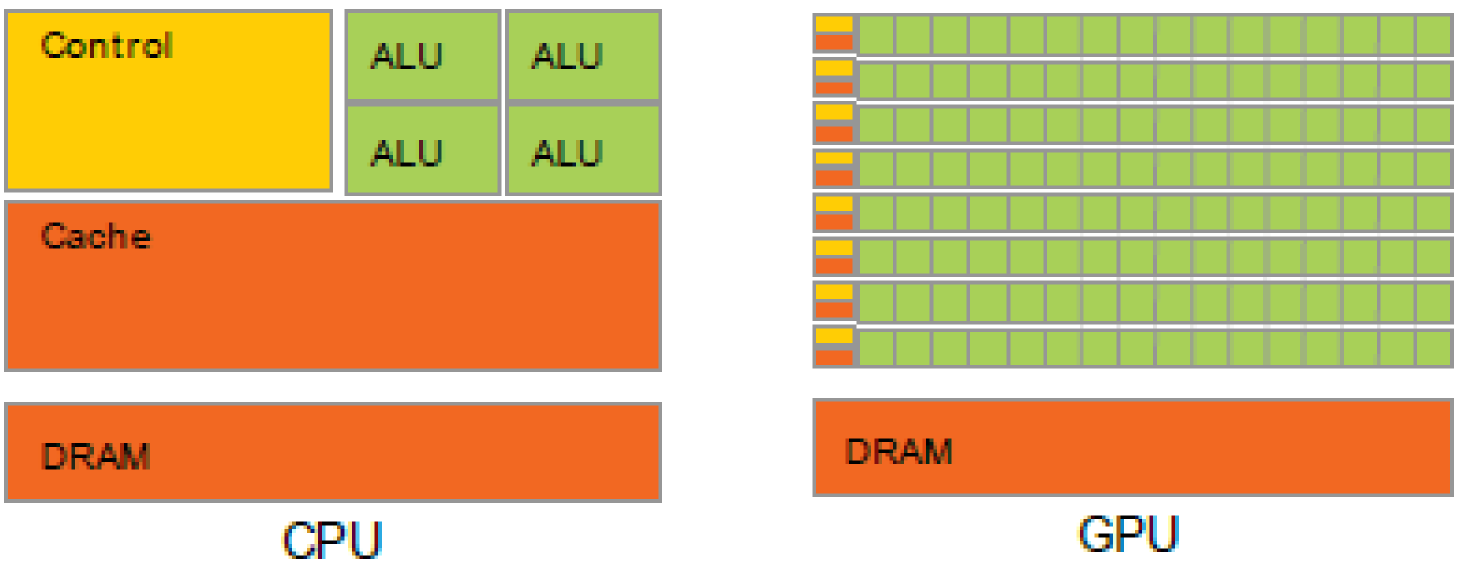
317 images
(hemisphere)

ground truth model

Goesele, Curless, Seitz, 2006

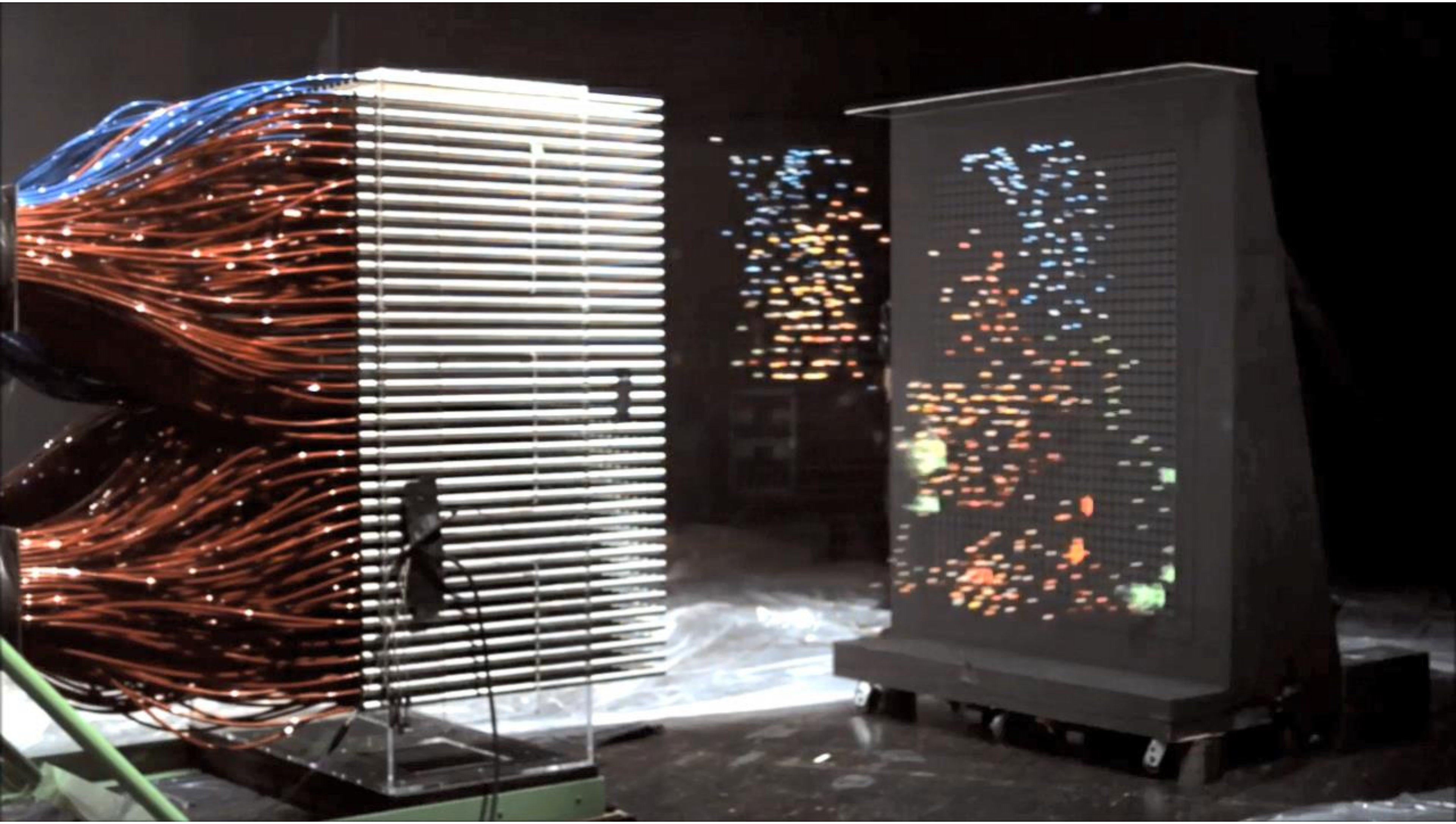
GPU: Graphics Processing Unit

- GPU performs calculations ***in parallel*** on thousands of cores (on a CPU a few cores optimized for *serial* processing)
- More transistors devoted to data processing
- More info: http://www.nvidia.com/object/what-is-gpu-computing.html#sthash.bW35IDmr.dpuf



ALU: Arithmetic Logic Unit

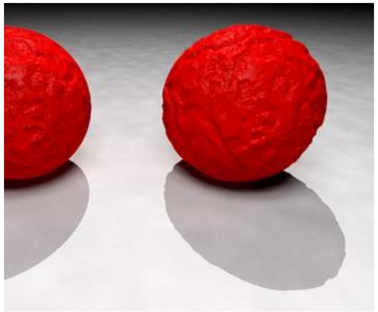
GPU: Graphics Processing Unit



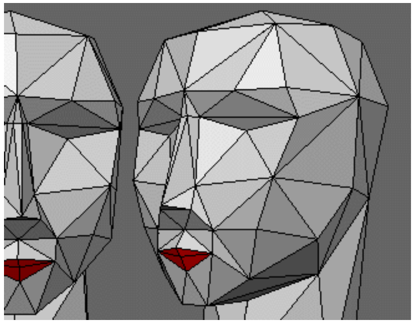
https://www.youtube.com/watch?v=-P28LKWTzrI

GPU Capabilities

- **Fast pixel processing**
  - Ray tracing, draw textures, shaded triangles faster than CPU
- **Fast matrix / vector operations**
  - Transform vertices
- **Deep Learning**



Bump mapping



Shaded triangles

GPU for 3D Dense Reconstruction

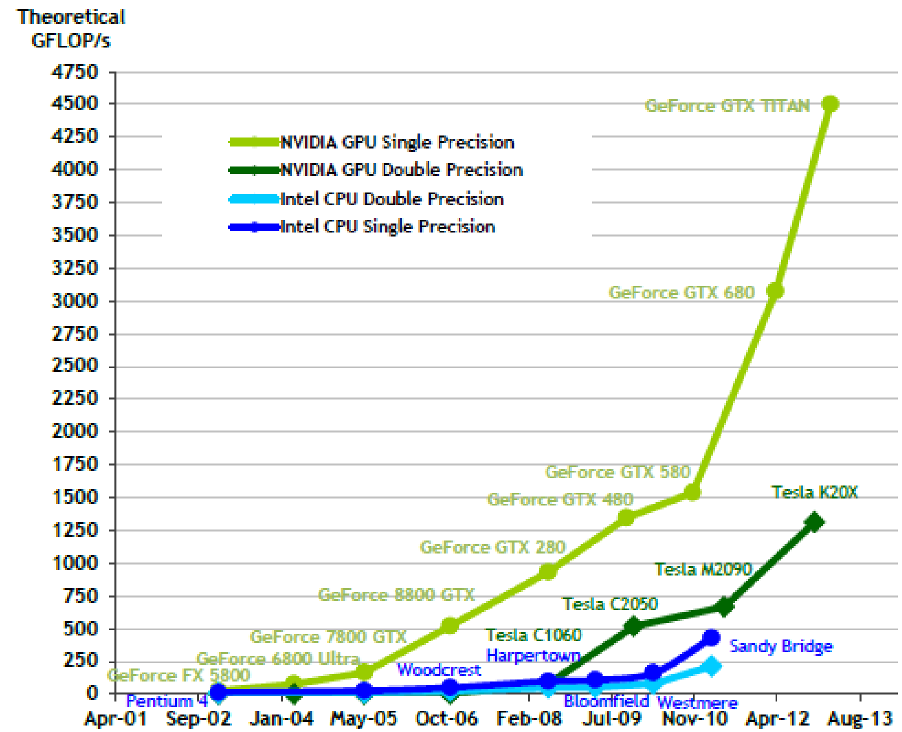
- **Image processing**
  - Filtering & Feature extraction (i.e., **convolutions**)
  - Warping (e.g., **epipolar rectification**, **homography**)

- **Multiple-view geometry**
  - Search for **dense correspondences**
    - *Pixel-wise* operations (SAD, **SSD**, NCC)
    - **Matrix and vector operations** (epipolar geometry)
  - **Aggregated Photometric Error** for multi-view stereo

- **Global optimization**
  - *Variational methods (i.e., regularization (smoothing))*
    - ***Parallel, in-place*** operations for gradient / divergence computation

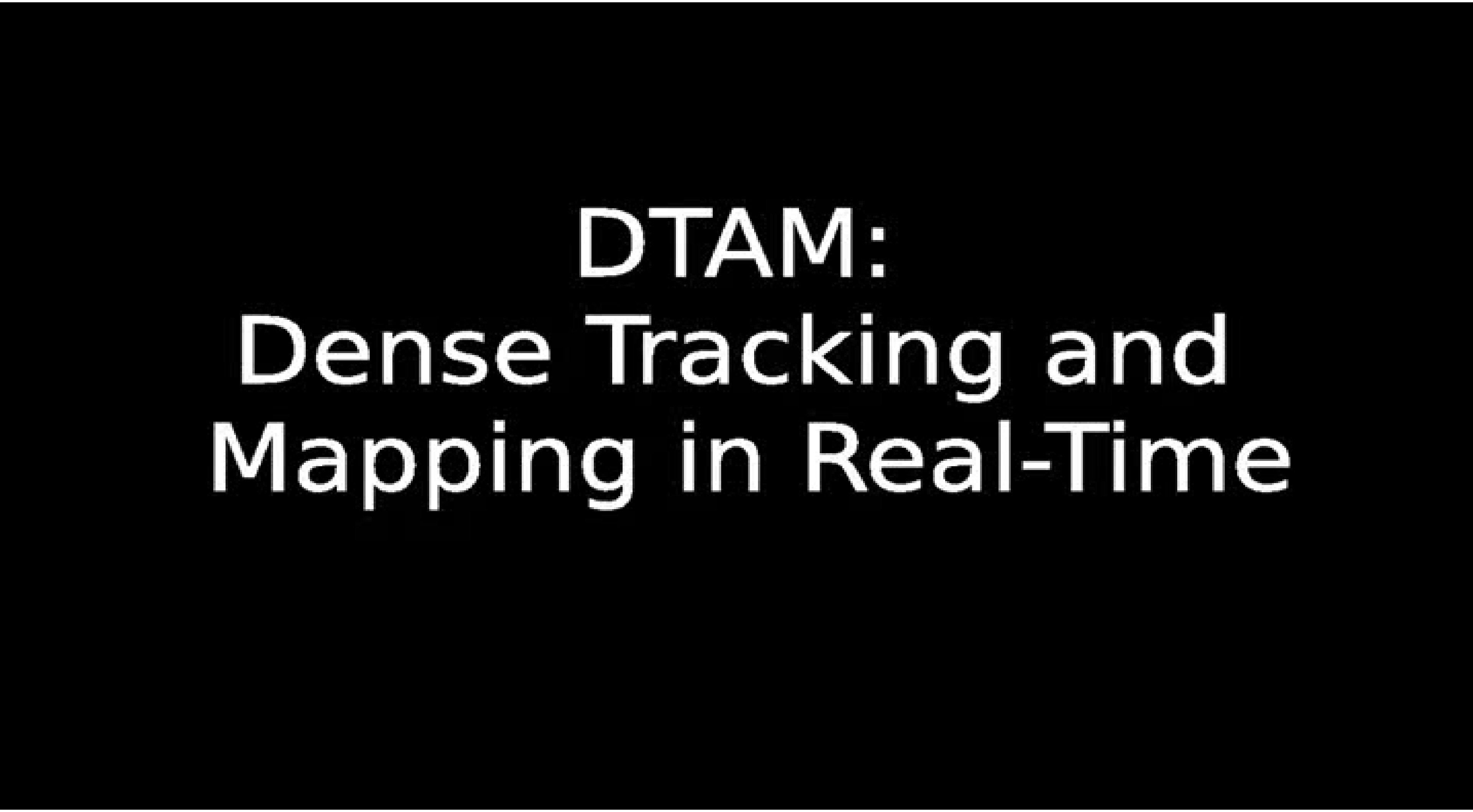
Why GPU

- GPUs run *thousands of lightweight threads* ***in parallel***
  - *Typically* on consumer hardware: 1000 threads per multiprocessor; 30 multiprocessor => **30k threads**.
  - Compared to CPU: 4 cores support 32 threads (with HyperThreading).
- Well suited for **data-parallelism**
  - The same instructions executed on multiple data in parallel
  - High **arithmetic intensity**: *arithmetic operations / memory operations*



[Source: nvidia]

DTAM: Dense Tracking and Mapping in Real-Time, ICCV'11 by Newcombe, Lovegrove, Davison



https://youtu.be/Df9WhgibCQA

REMODE:
Regularized Monocular Dense Reconstruction

[M. Pizzoli, C. Forster, D. Scaramuzza, REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time, IEEE International Conference on Robotics and Automation 2014]

Open source: *https://github.com/uzh-rpg/rpg_open_remode*

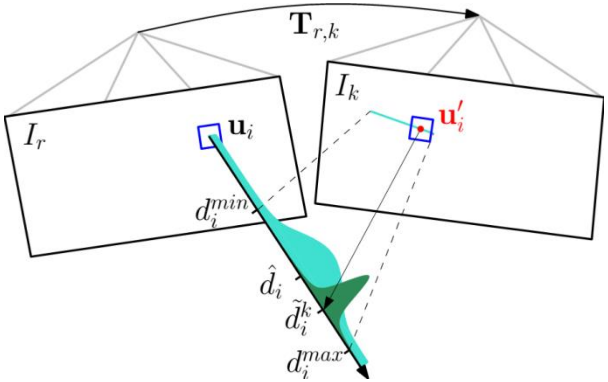

Displays
Global Options
Fixed Frame
/world
Background Color
255; 255; 255
Global Status: Ok
Add
Image
Image

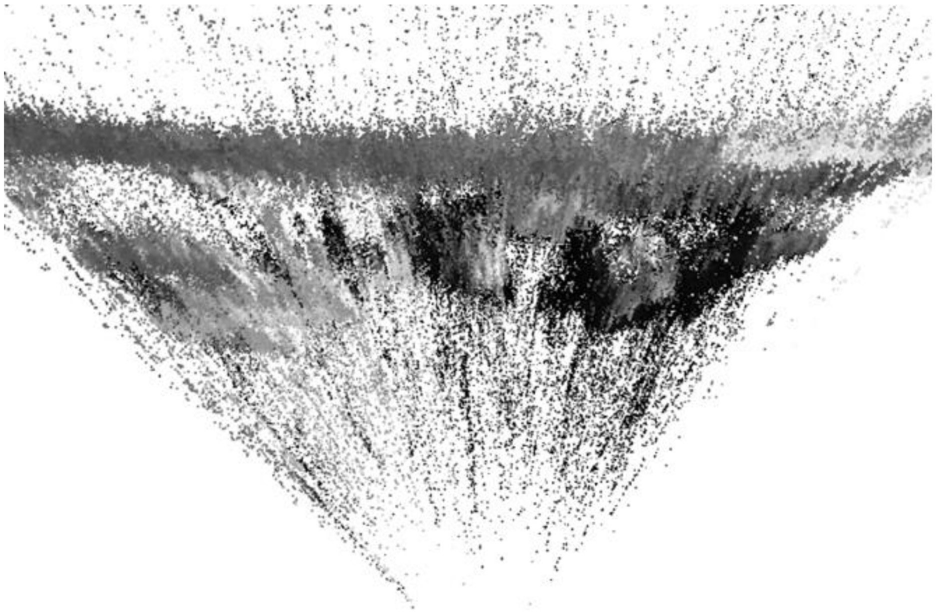
Monocular dense reconstruction
in real-time from a hand-held camera

Stage-set from Gruber et al., "The City of Sights", ISMAR'10.

REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time

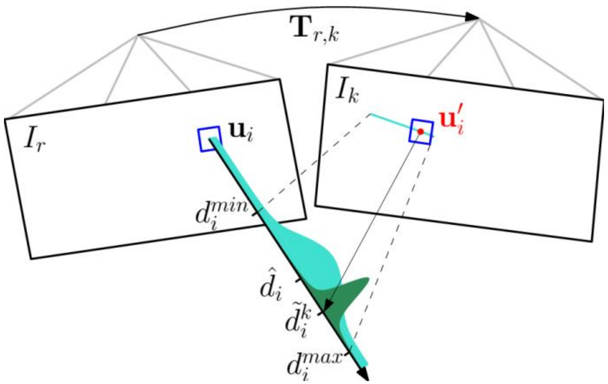
- Tracks every pixel (like DTAM) but **probabilistically** via recursive Bayesian estimation
- Runs live on video streamed from MAV (50 Hz on GPU)
- **Regularizes only** 3D points with **low depth uncertainty**
  - does not fill holes, if present.
    Great for robotic applications!





REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time

- Tracks every pixel (like DTAM) but **probabilistically** via recursive Bayesian estimation
- Runs live on video streamed from MAV (50 Hz on GPU)
- **Regularizes only** 3D points with **low depth uncertainty**
  - does not fill holes, if present.
    Great for robotic applications!

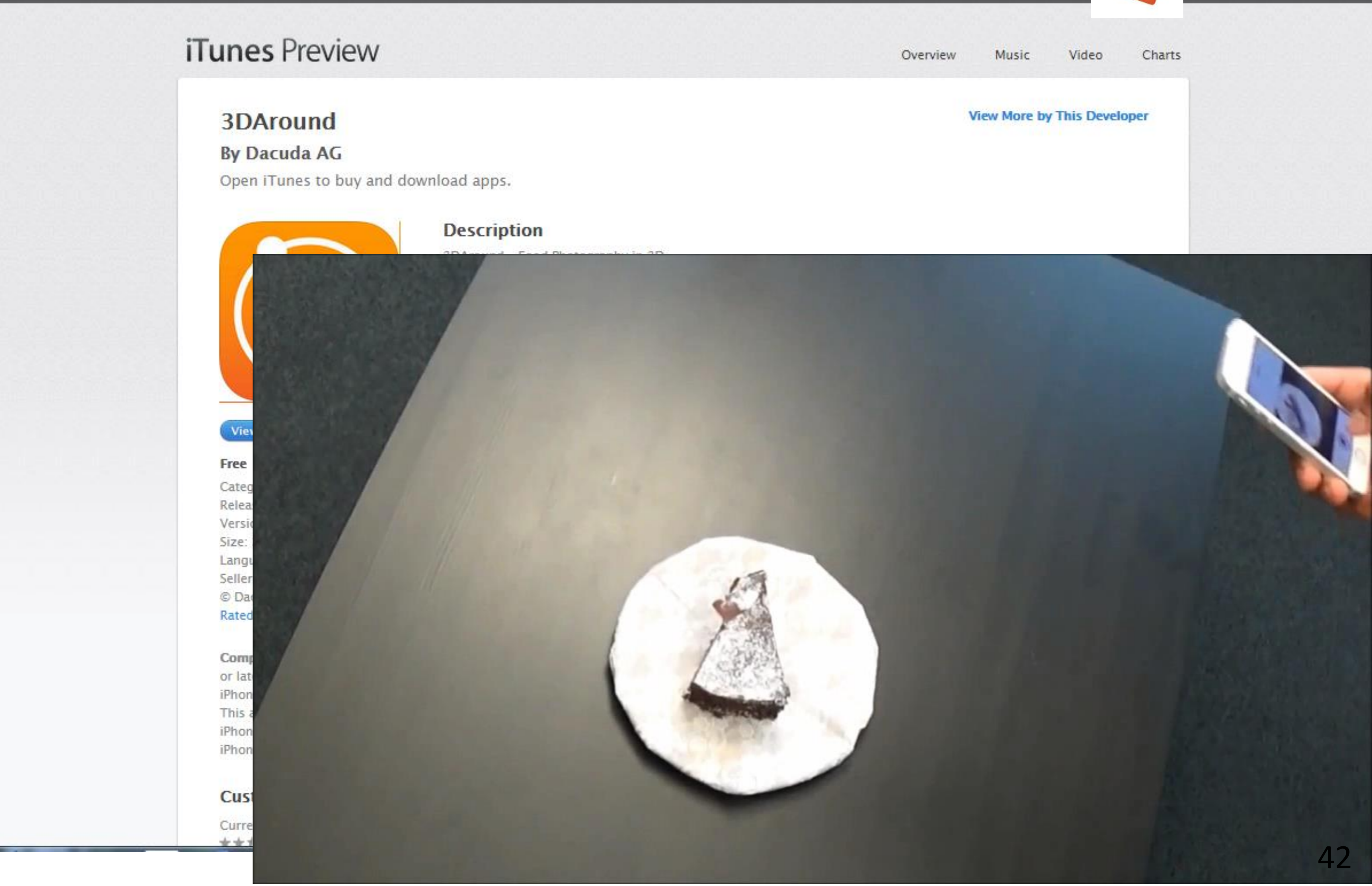


REMODE applied to autonomous flying 3D scanning

Live demonstration at the Firefighter Training Area of Zurich

3DAround iPhone App

Dacuda



DynamicFusion

- Simultaneous Reconstruction of non-rigid scenes and 6-DOF camera pose tracking using an RGBD camera

Live Input Depth Map

Live Model Output

Live RGB Image (unused)

Canonical Model Reconstruction

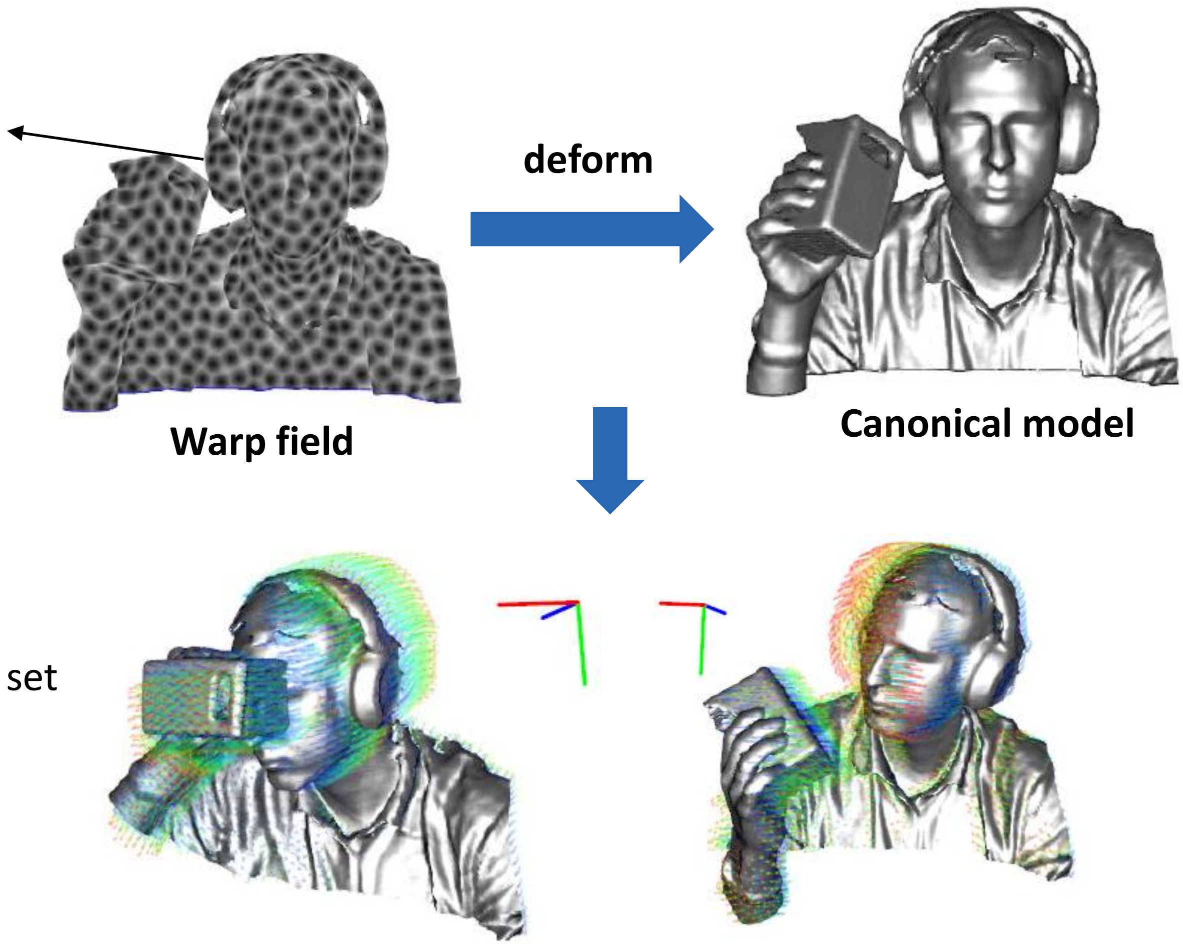
Warped Model

Newcombe et.al. DynamicFusion: Reconstruction and Tracking of Non-rigid Scenes in Real-Time. CVPR 2015, Best Paper Award.

DynamicFusion: scene representation

- How to represent the deformation of the scene?
  → Dense warp field

Each node stands for a rigid body motion that transforms (locally) the canonical (static) model to the current, live frame.



We need to estimate a set of sparse nodes in the warp field <u>per frame</u>.

Live Frames: warped model

Newcombe et.al. DynamicFusion: Reconstruction and Tracking of Non-rigid Scenes in Real-Time.

DynamicFusion: tracking and model update

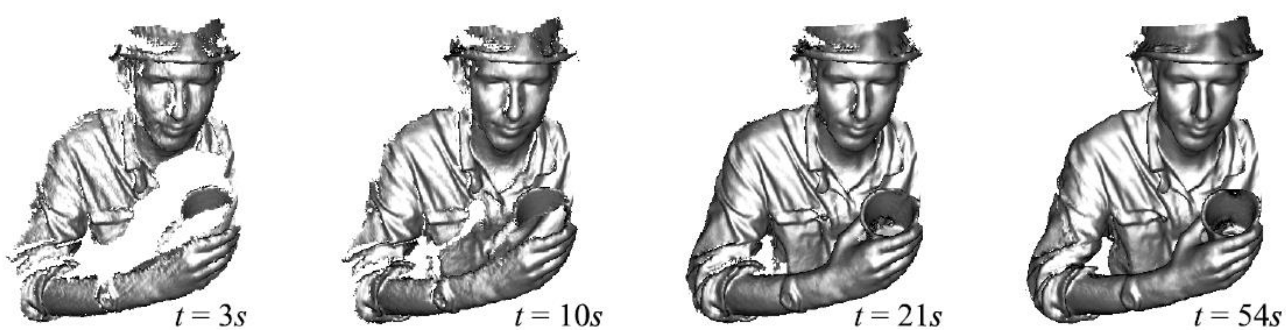
- Tracking: many parameters to optimize
  - Camera motion
  - The nodes in the warp field

W_t: warp field
D_t: depth map
V: canonical model

$$E(\mathcal{W}_t, \mathcal{V}, D_t, \mathcal{E}) = \mathbf{Data}(\mathcal{W}_t, \mathcal{V}, D_t) + \lambda \mathbf{Reg}(\mathcal{W}_t, \mathcal{E})$$

- **Data** term: The warped model should agree well with the depth map.
- **Regularization** term: The warp field should be smooth.

- Model update: update the canonical model recursively
  → does not need to store all the depth images



Things to remember

- Aggregated Photometric Error
- Disparity Space Image
- Effects of regularization
- Handling discontinuities
- GPU

- Readings:
  - Chapter: 11.6 of Szeliski's book

Understanding Check

Are you able to answer the following questions?

- Are you able to describe the multi-view stereo working principle? (aggregated photometric error)
- What are the differences in the behavior of the aggregated photometric error for corners, flat regions, and edges?
- What is the disparity space image (DSI) and how is it built in practice?
- How do we extract the depth from the DSI?
- How do we enforce smoothness (regularization) and how do we incorporate depth discontinuities (mathematical expressions)?
- What happens if we increase lambda (the regularization term)? What if lambda is 0? And if lambda is too big?
- What is the optimal baseline for multi-view stereo?
- What are the advantages of GPUs?