University of Zurich[UZH] | ETH zürich

Institute of Informatics – Institute of Neuroinformatics

ROBOTICS & PERCEPTION GROUP

Lecture 10
Dense 3D Reconstruction

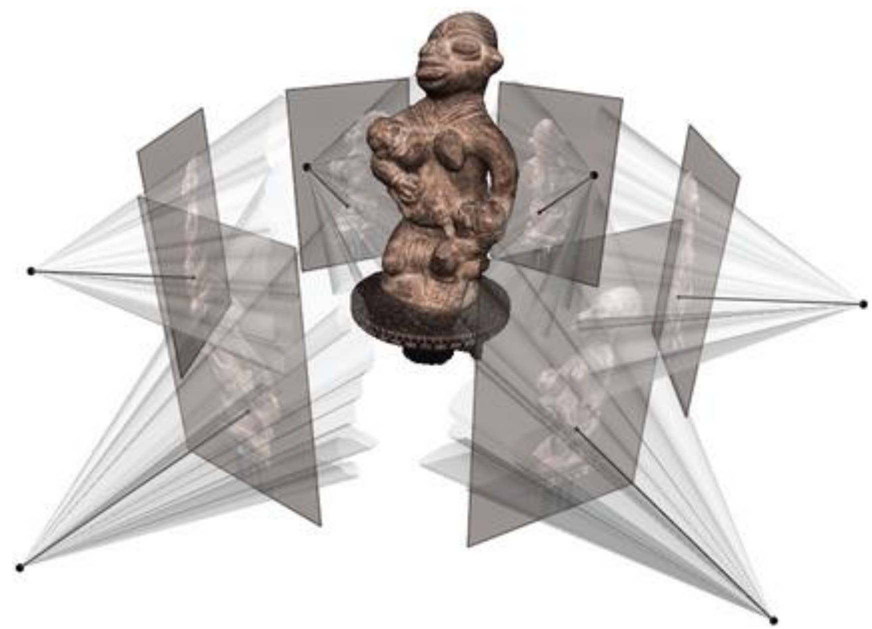
Davide Scaramuzza

REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time

M. Pizzoli, C. Forster, D. Scaramuzza, REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time, ICRA'14. [PDF]

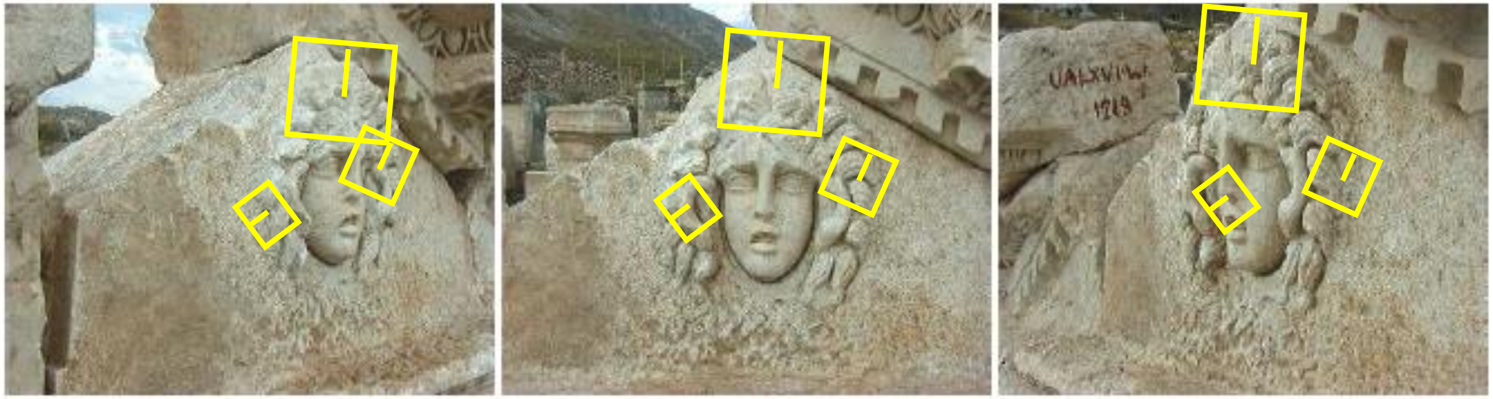
Dense Reconstruction (or Multi-view stereo)

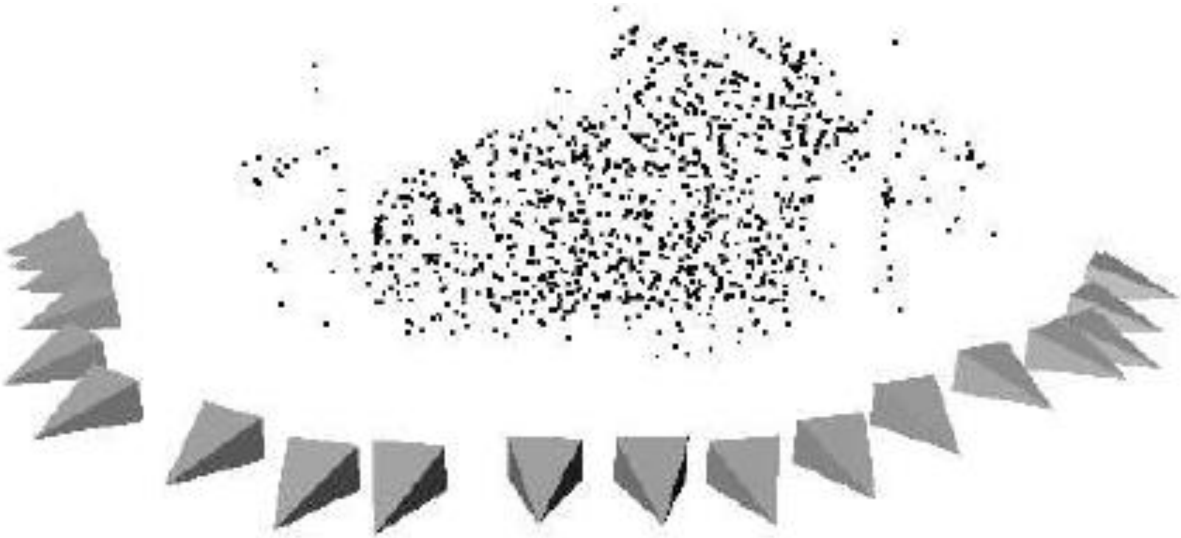
- **Input**: calibrated images from several viewpoints (i.e., K, R, T are known for each camera, e.g., from SFM)
- **Output**: 3D object **dense** reconstruction



Sparse Reconstruction

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





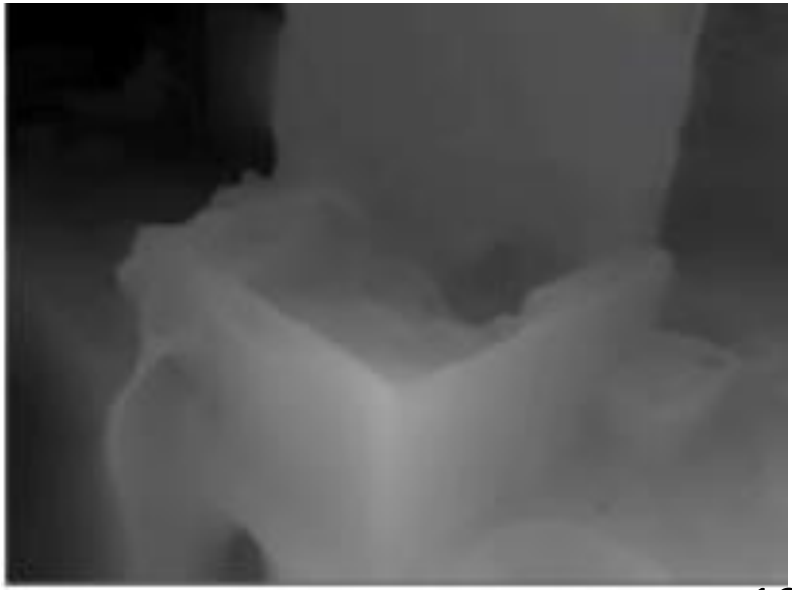
Dense Reconstruction

- Estimate the structure from a "dense" region of pixels

Dense reconstruction workflow

Step 1: Local methods

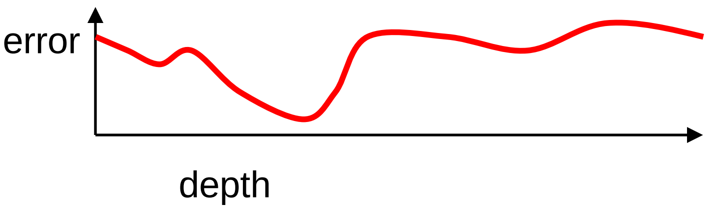
– Estimate depth for every pixel independently (how do we compute correspondences for *every* pixel?)



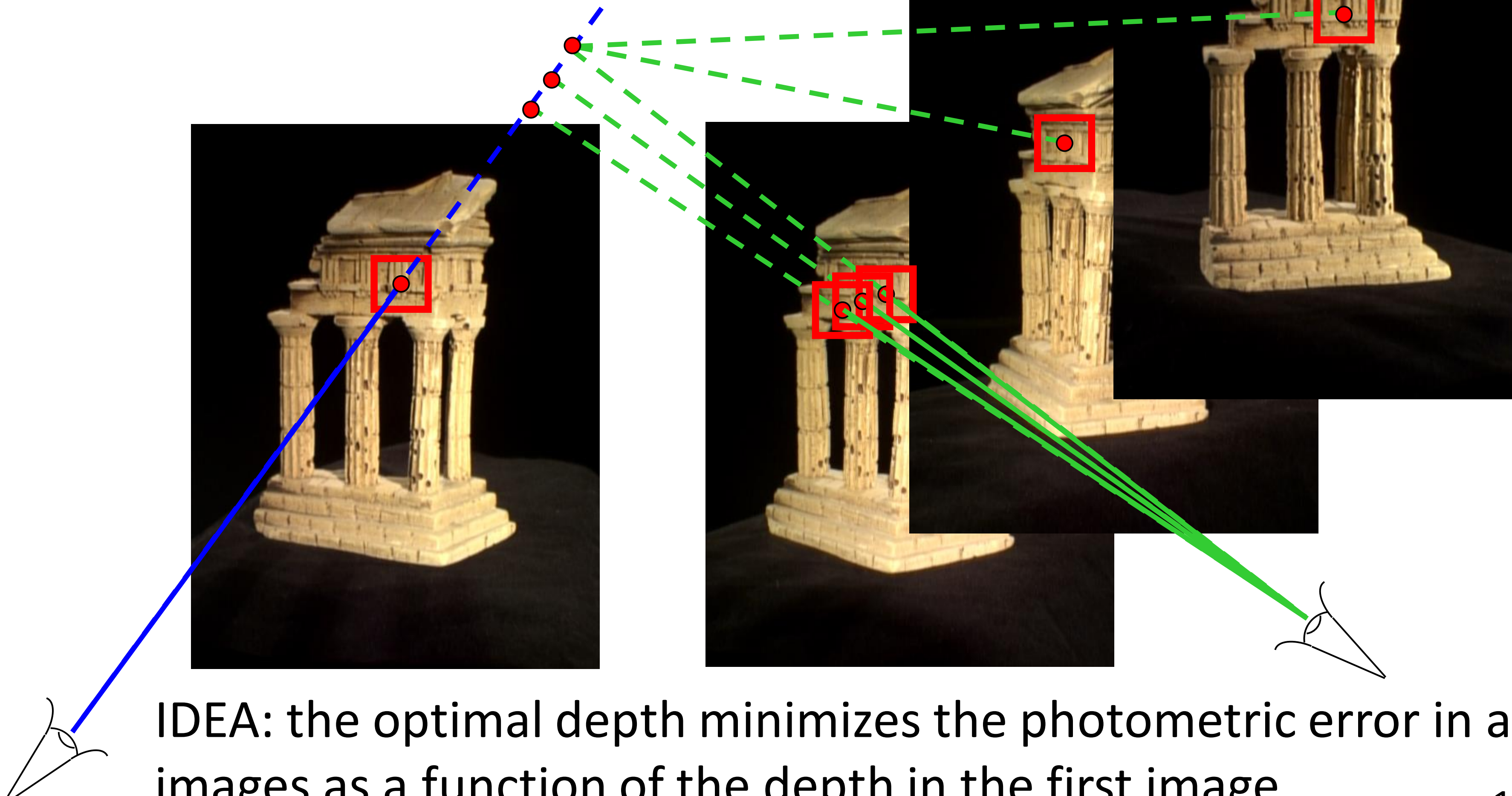
Step 2: Global methods

– Refine the depth surface as a whole by enforcing smoothness constraint

Photometric error (SSD)



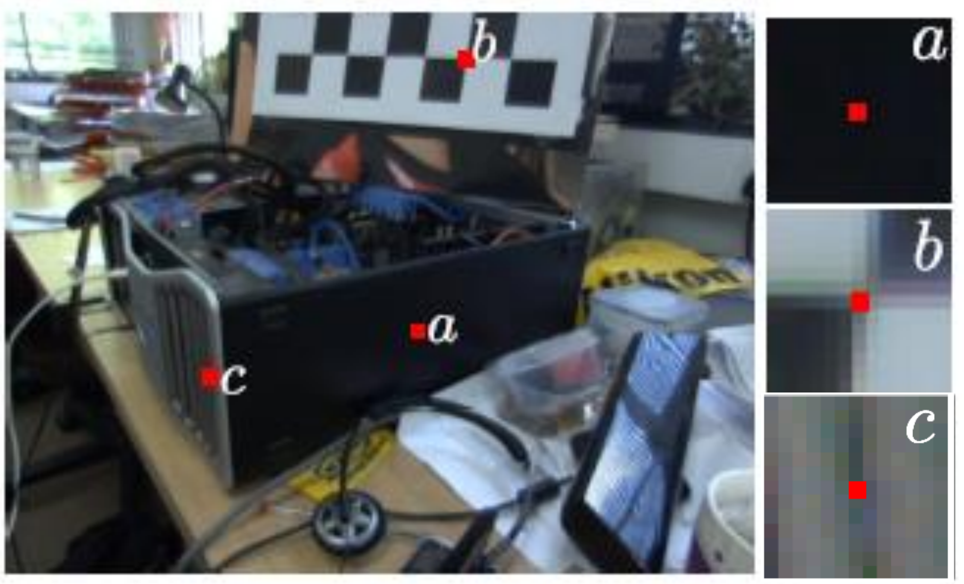
This error plot is derived for every combination of the reference image and any further image



IDEA: the optimal depth minimizes the photometric error in all images as a function of the depth in the first image

Aggregated Photometric Error

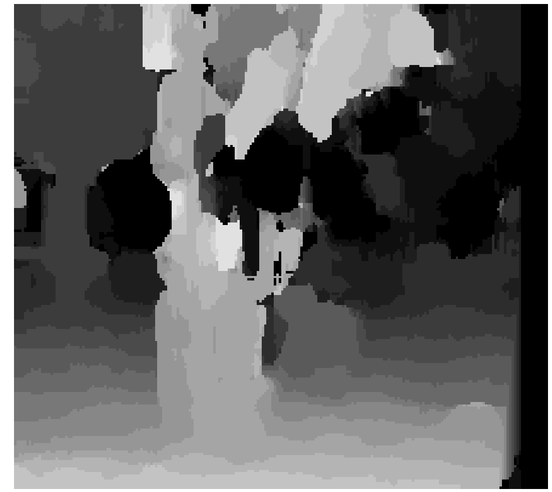
- Dense reconstruction requires establishing dense correspondences
- Correspondences are computed based on photometric error:
  - SSD between corresponding patches of intensity values (min patch size: 1x1 pixels)
  - What are the pros and cons of using small or large patches? (recall from stereo: see next slide)
- Not all the pixels can be matched reliably
  - Viewpoint and illumination changes, occlusions
- Take advantage of many small-baseline views where high quality matching is possible (why?)



[Newcombe et al. 2011]

Review from stereo vision: Effects of window size

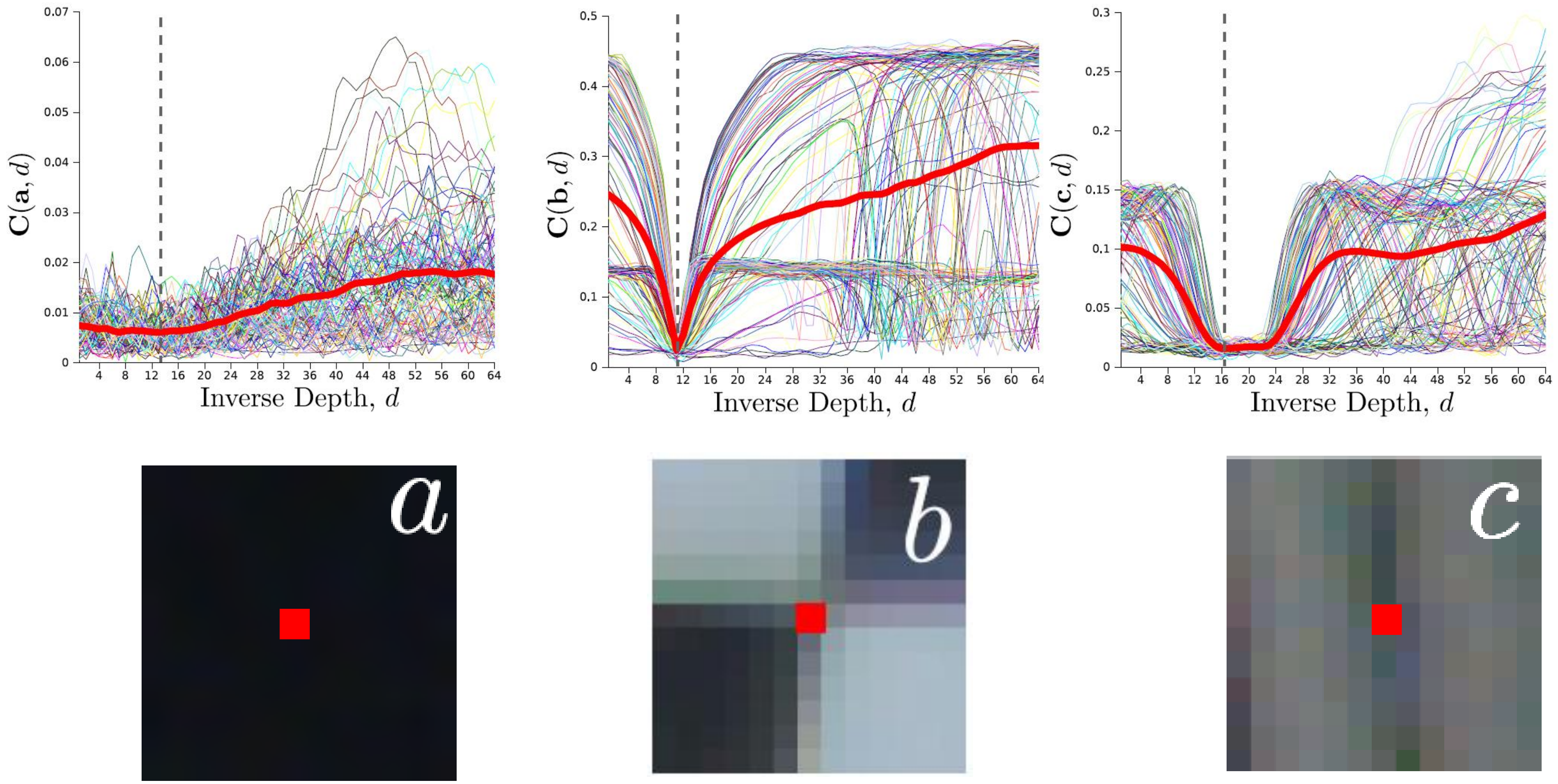
W = 3



W = 20

- Smaller window
  - \+ More detail
  - – More noise
- Larger window
  - \+ Smoother disparity maps
  - – Less detail

Aggregated Photometric Error



- 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

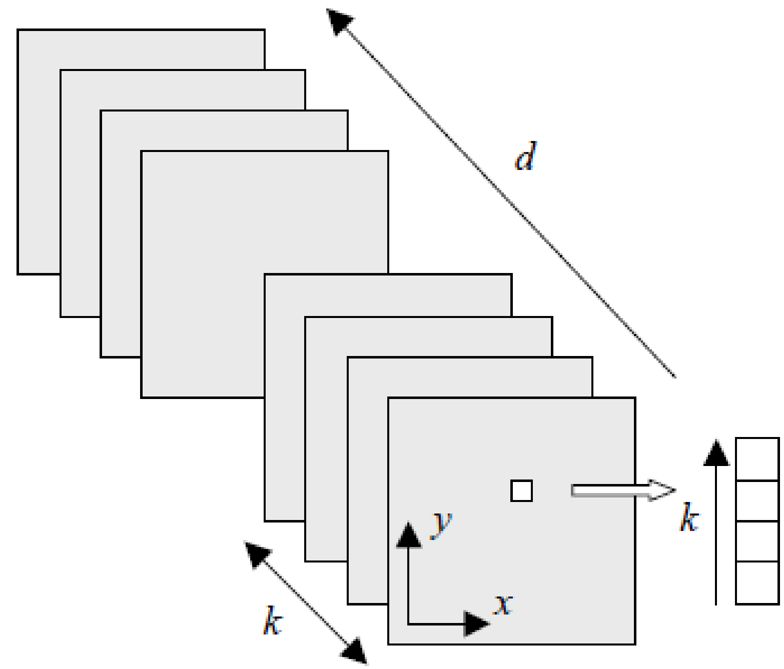
Disparity Space Image (DSI)

- For a given image point (u, v) and for discrete depth hypotheses d, the aggregate 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:

$$C(u, v, d) = \sum_k \rho(\widetilde{I_k}(u', v', d) - I_r(u, v))$$

where $\widetilde{I_k}(u', v', d)$ 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

- $\rho(\cdot)$ is the photometric error (SSD)



[Szeliski and Golland 1999]

Disparity Space Image (DSI)

240 x 180 x 100 voxels

Solution

The solution to the depth estimation problem *is a* **function** $\boldsymbol{d}(\boldsymbol{u}, \boldsymbol{v})$ in the DSI that satisfies two criteria:

Minimum aggregated photometric error (i.e., $arg \min_{d} C$)

AND

Piecewise smooth (global methods)

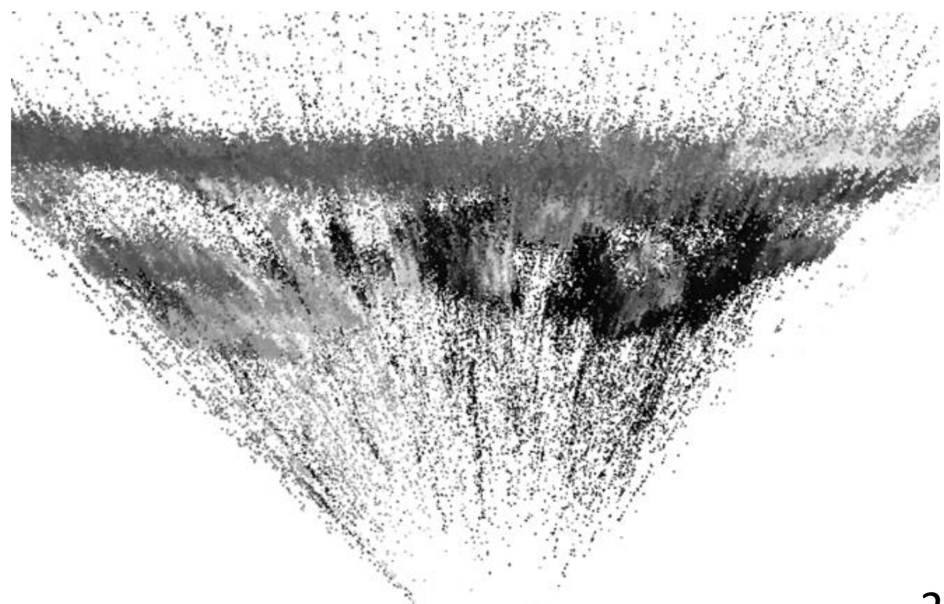
Solution

The solution to the depth estimation problem *is a* **function $\boldsymbol{d}(\boldsymbol{u}, \boldsymbol{v})$** in the DSI that satisfies two criteria:

Minimum aggregated photometric error (i.e., $arg \min_{d} C$)

AND

Piecewise smooth (global methods)



Solution

The solution to the depth estimation problem *is a* **function $\boldsymbol{d}(\boldsymbol{u},\boldsymbol{v})$** in the DSI that satisfies two criteria:

Minimum aggregated photometric error (i.e., $arg\min_{d} C$)

AND

Piecewise smooth (global methods)

Solution

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

$$E(d) = \underbrace{E_d(d)}_{\text{Data term}} + \underbrace{\lambda E_s(d)}_{\text{Regularization term}}$$

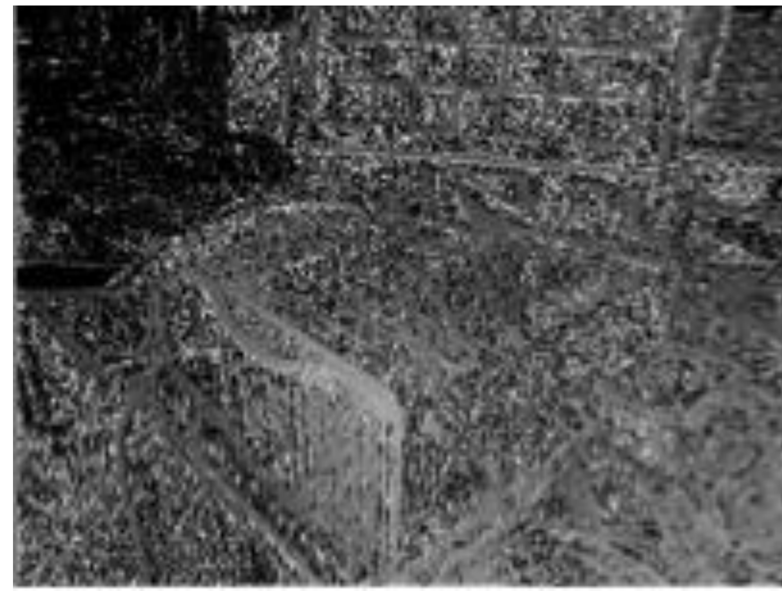
$$E_d(d) = \sum_{(u,v)} C(u, v, d(u, v))$$

$$E_s(d) = \sum_{(u,v)} \rho_d\big(d(u, v) - d(u + 1, v)\big) + \rho_d\big(d(u, v) - d(u, v + 1)\big)$$

  - ρ_d is a norm (e.g. L_2, L_1 or Huber norm)
  - 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) as well as discontinuities
  - ***Fills the holes***



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

Regularized depth maps

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

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

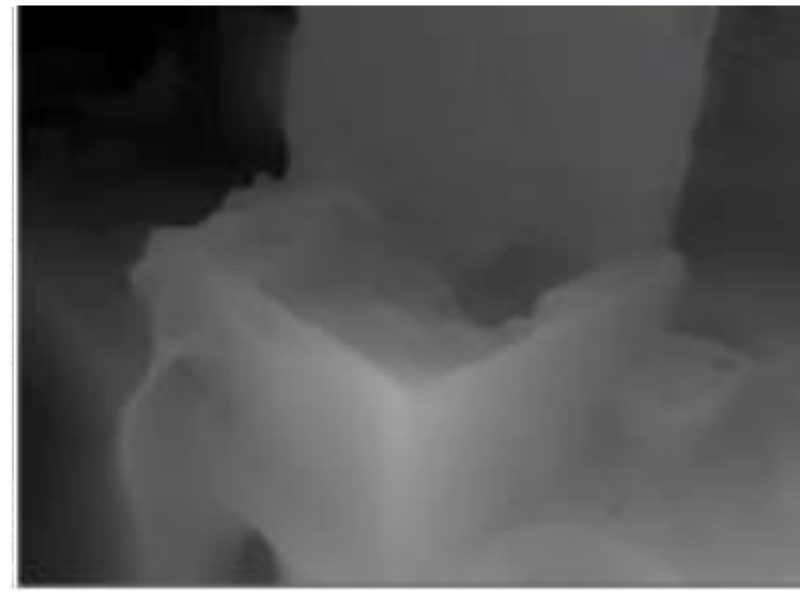
Regularized depth maps

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

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

Regularized depth maps

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



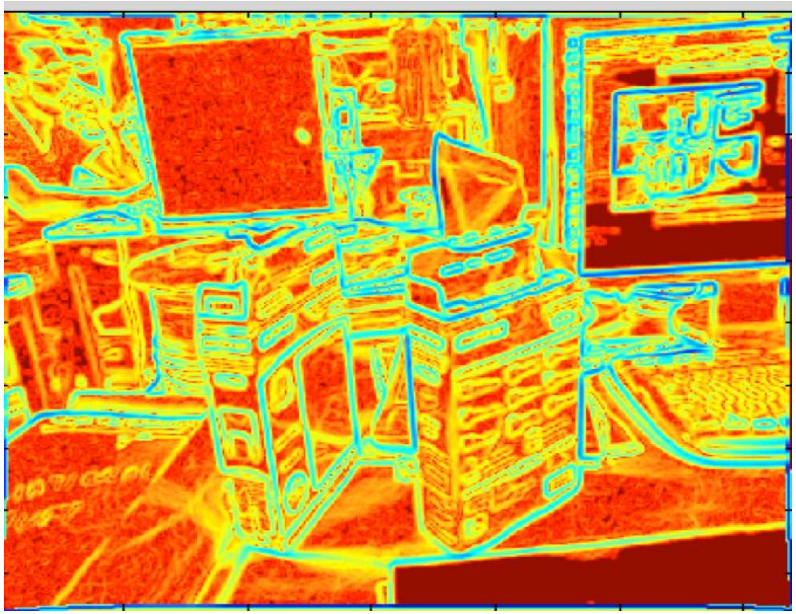
Final depth image for different λ
[Newcombe et al. 2011]

Regularized depth maps

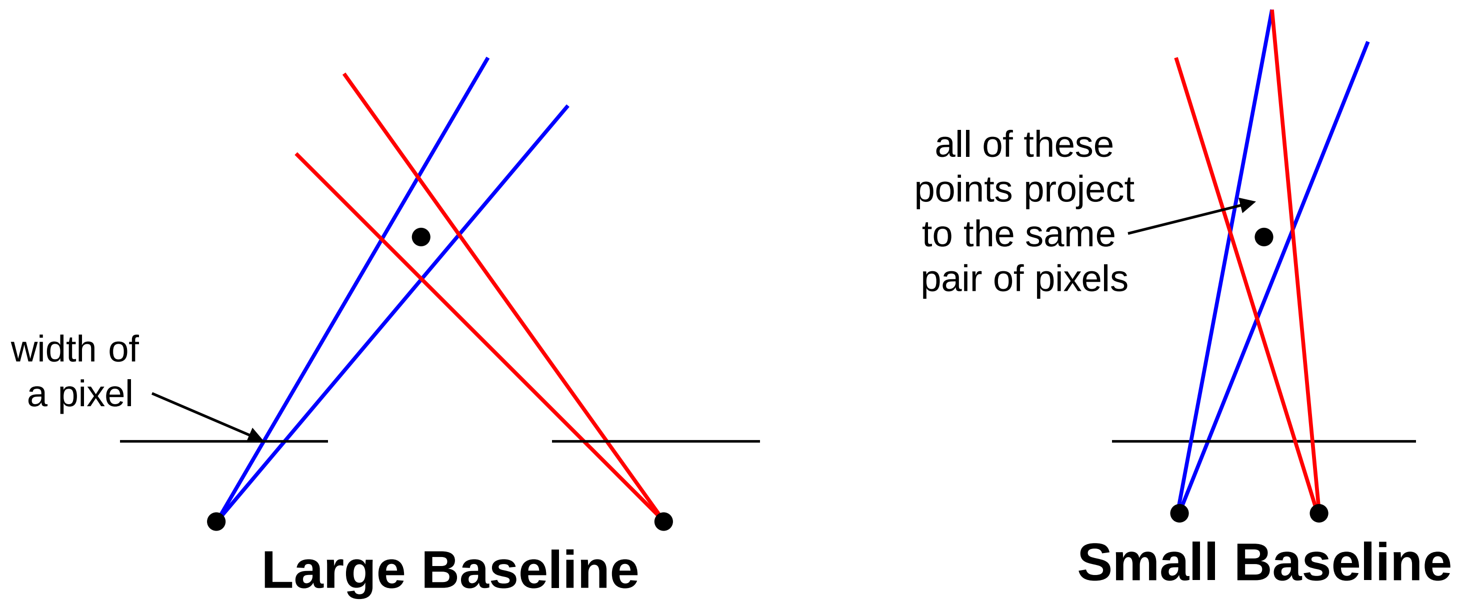
- Popular assumption: *discontinuities in intensity* **coincide** with *discontinuities in depth*
- ***Control smoothness penalties*** according to image gradient

$$\rho_d\big(d(u,v) - d(u+1,v)\big) \cdot \rho_I\,(\|I(u,v) - I(u+1,v)\|)$$

- ρ_I is some *monotonically decreasing* function of intensity differences: ***lower*** *smoothness cost for* ***high intensity gradients***



Choosing the stereo baseline



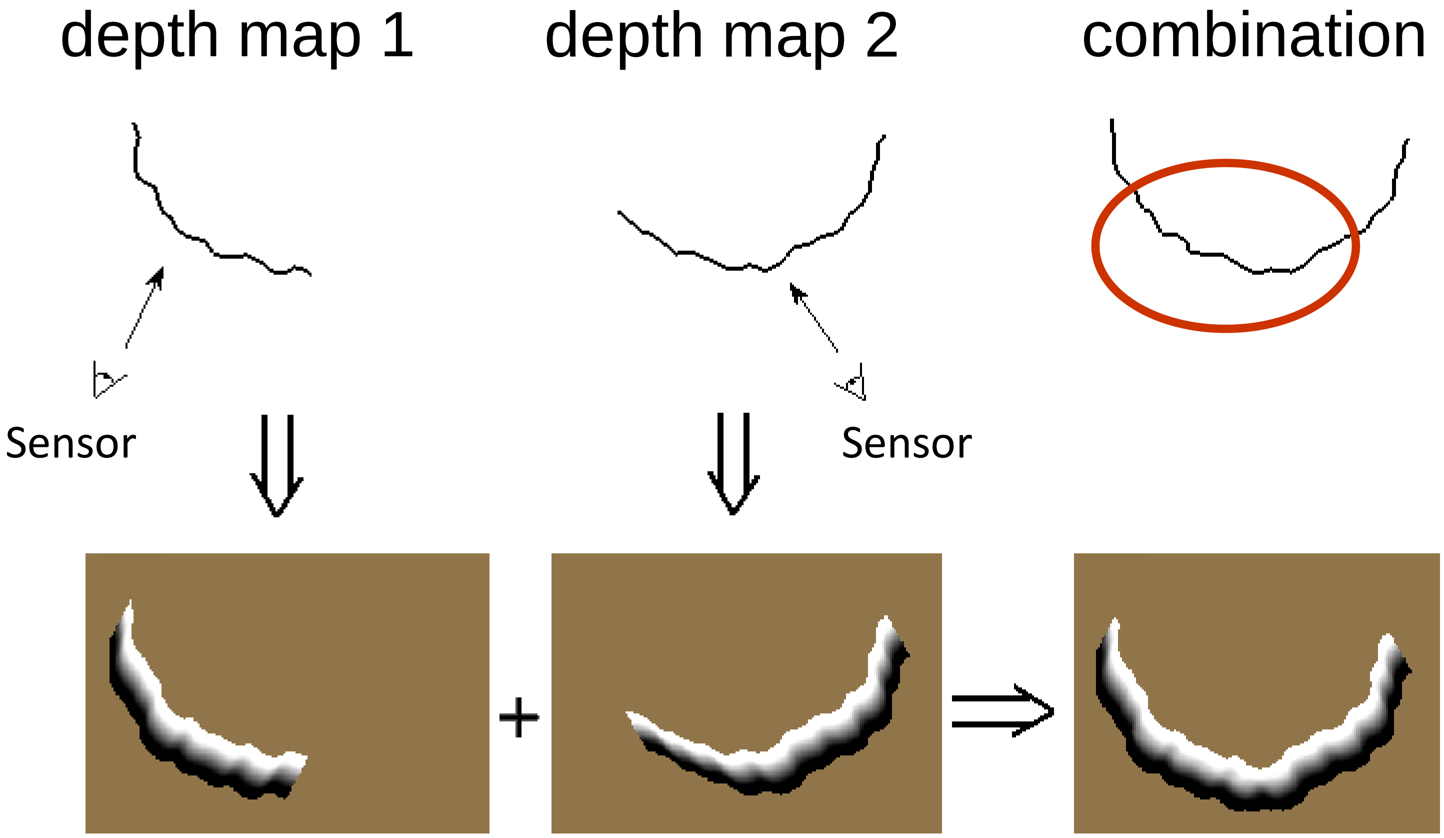
What's the optimal baseline ?

- Too small: ***large depth error***
- Too large: ***difficult search problem***

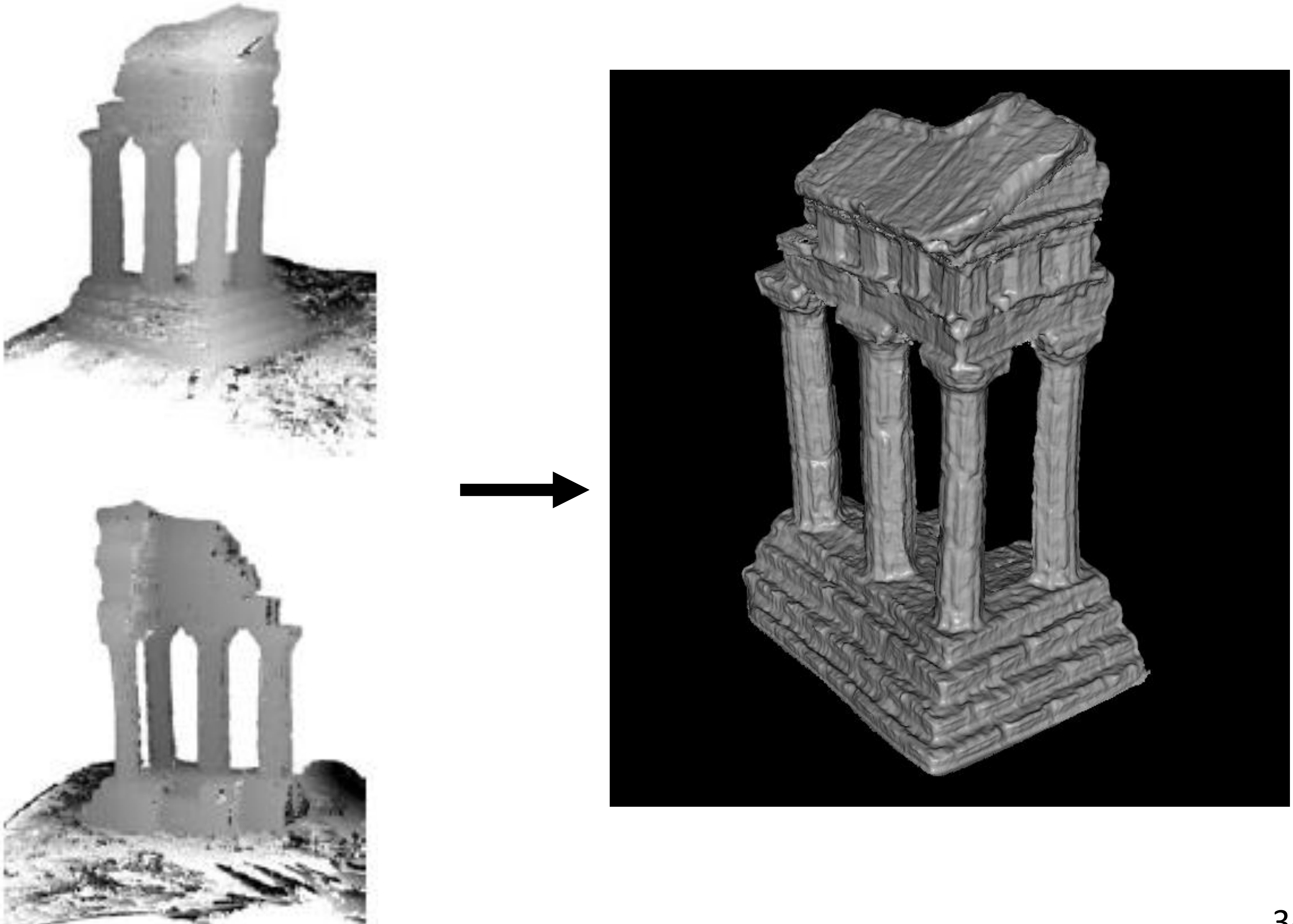
Solution

- Obtain depth map from **small baselines**
- When baseline becomes too large, **create new reference frame** (keyframe) and start a new depth computation

Fusion of multiple depth maps



Fusion of multiple depth maps



Depth map fusion

input image

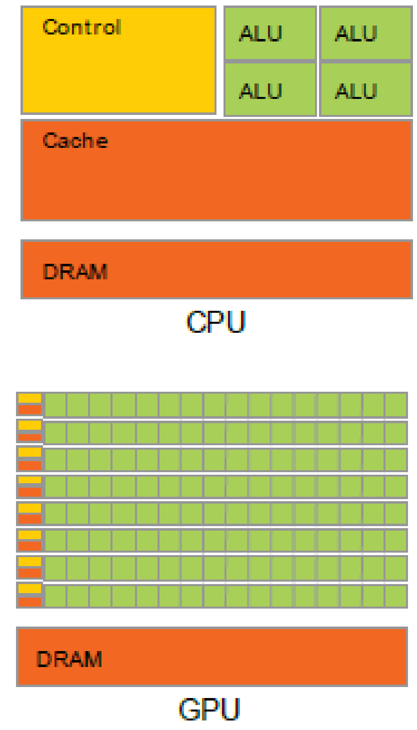
317 images
(hemisphere)

ground truth model

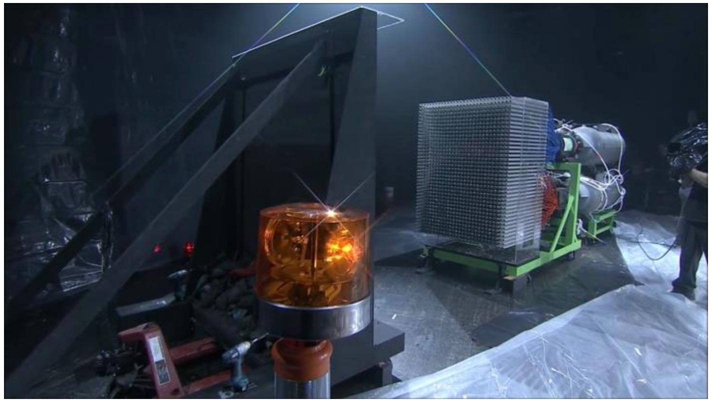
[Goesele, Curless, Seitz, 2006]

GPGPU

ALU: Arithmetic Logic Unit



- ***GPGPU = G**eneral **P**urpose computing on **G**raphics **P**rocessing **U**nit*
- Perform demanding calculations on the GPU instead of the CPU
- On the GPU: high processing power ***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

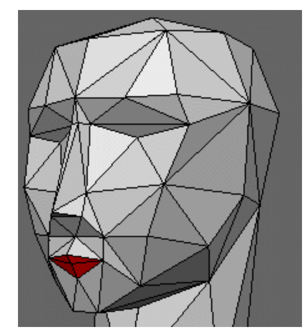


https://www.youtube.com/watch?v=-P28LKWTzrI

GPU Capabilities

- **Fast pixel processing**
  - Ray tracing, draw textures, shaded triangles faster than CPU
- **Fast matrix / vector operations**
  - Transform vertices
- **Programmable**
  - Shading, bump mapping
- **Floating-point support**
  - Accurate computations
- Deep Learning

Bump mapping



Shaded triangles

GPU for 3D Dense Reconstruction

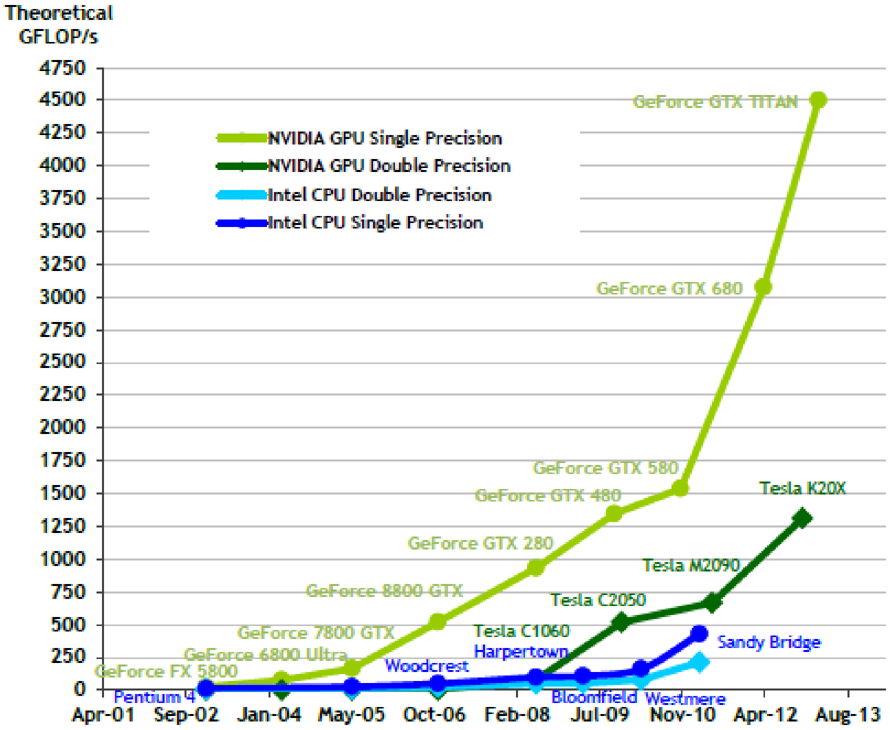
- **Image processing**
  - Filtering & Feature extraction (i.e., convolutions)
  - Warping (e.g., epipolar rectification, homography)

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

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

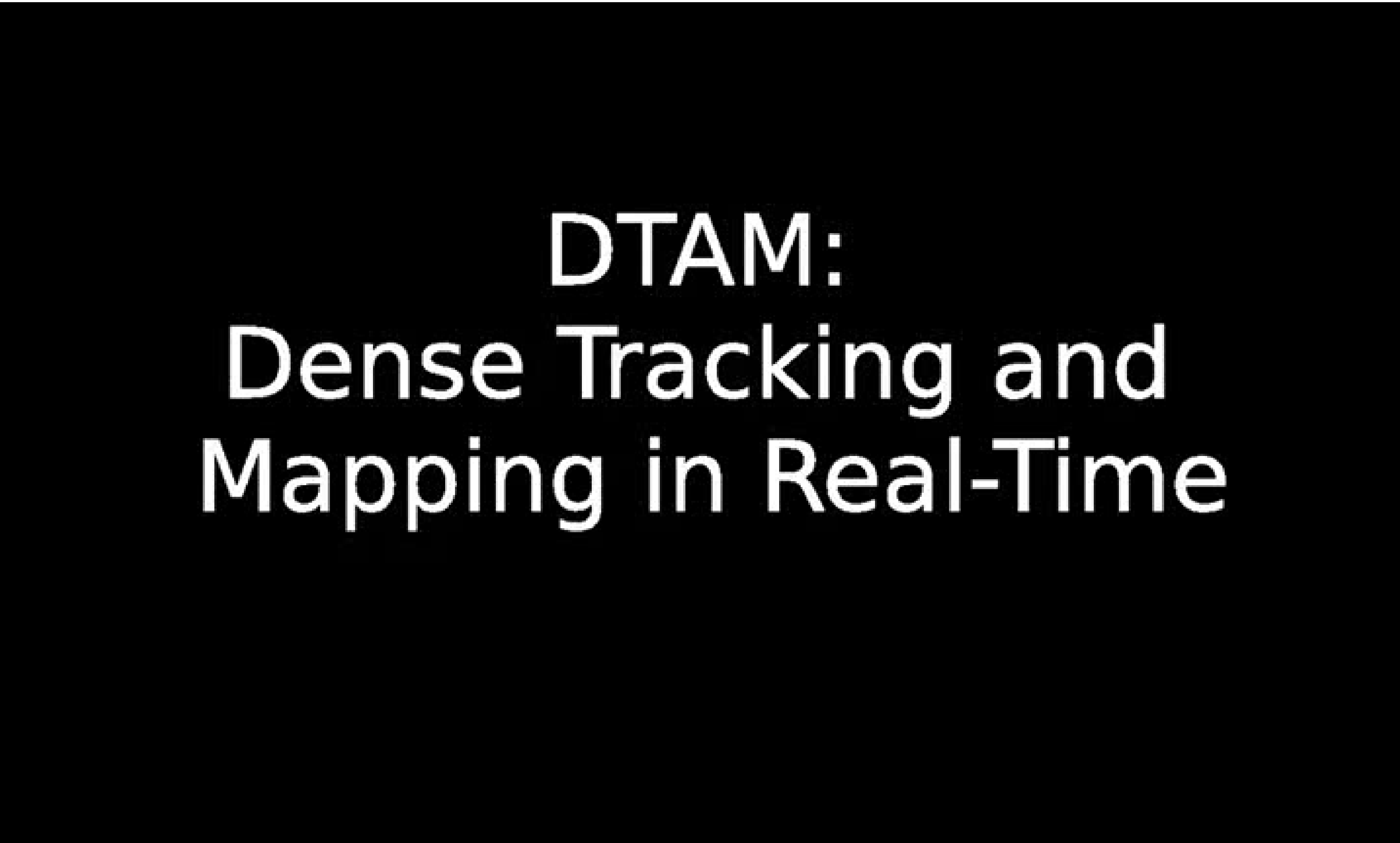
Why GPU

- GPUs run *thousands of lightweight threads* ***in parallel***
  - *Typically* on consumer hardware: 1024 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

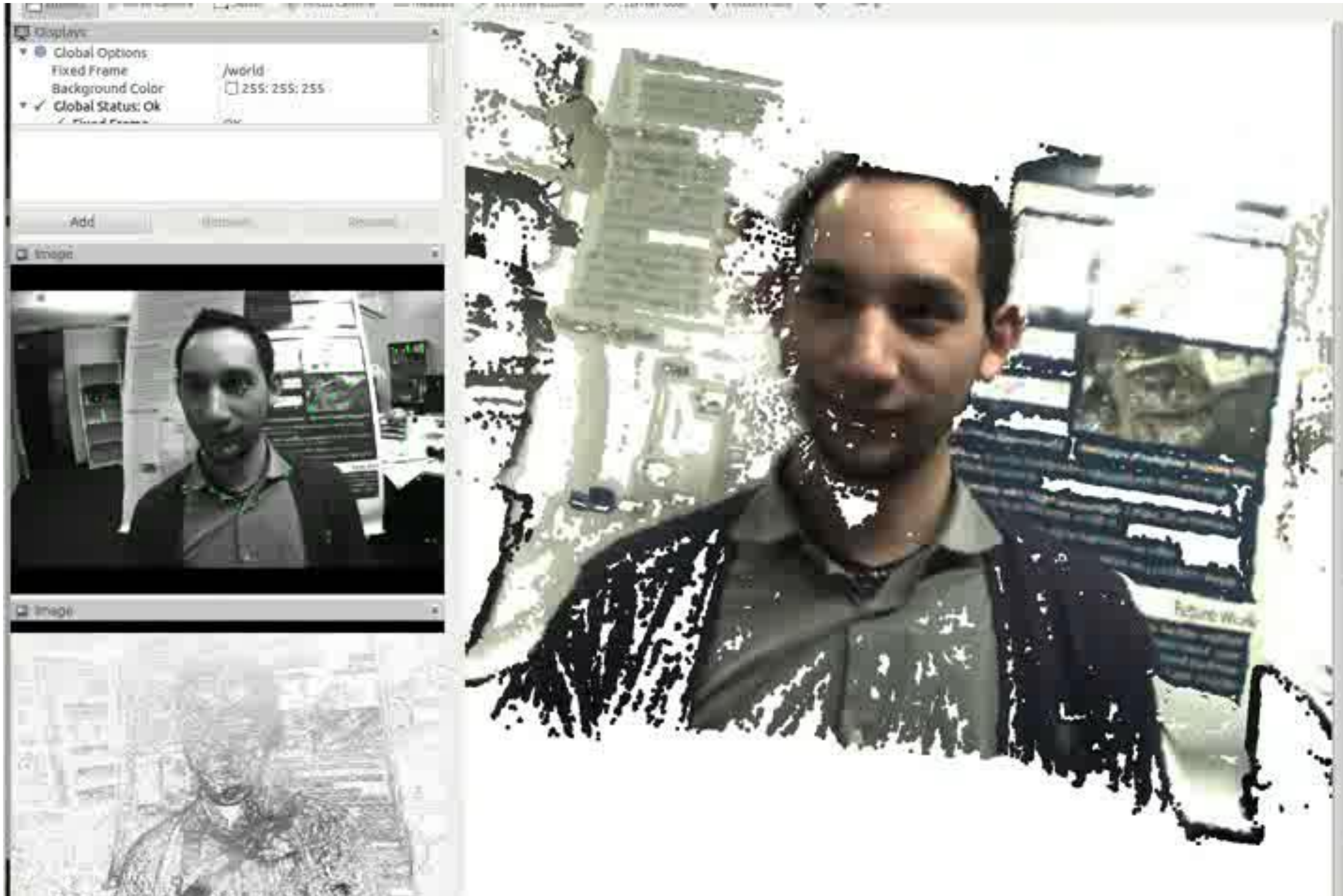


REMODE:
Regularized Monocular Dense Reconstruction

[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*

REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time, ICRA'14, by Pizzoli, Forster, Scaramuzza



REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time, ICRA’14, by Pizzoli, Forster, Scaramuzza

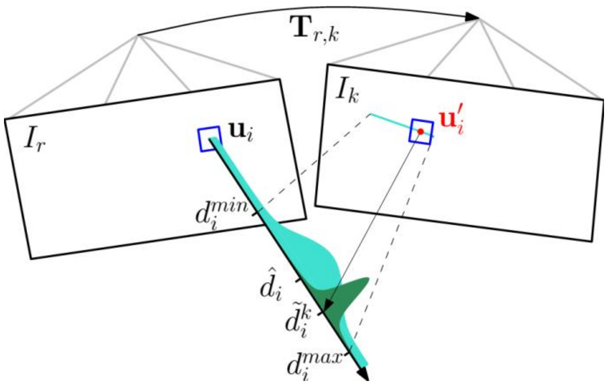
Monocular dense reconstruction
in real-time from a hand-held camera

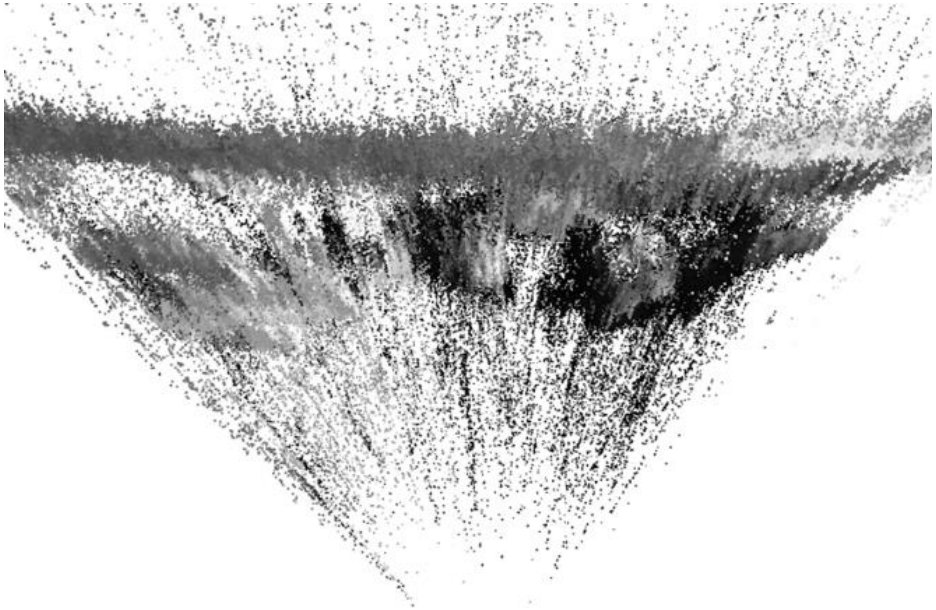
Stage-set from Gruber et al., "The City of Sights", ISMAR'10.

[M. Pizzoli, C. Forster, D. Scaramuzza, REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time, ICRA’14]

REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time, ICRA'14, by Pizzoli, Forster, Scaramuzza

- Tracks every pixel (like DTAM) but **Probabilistically**
- Runs live on video streamed from MAV (50 Hz on GPU)
- Copes well with low texture surfaces

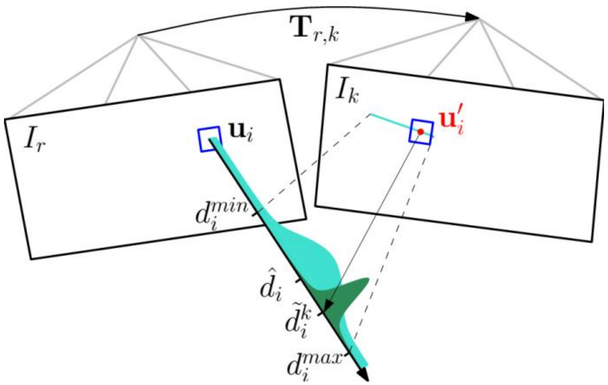


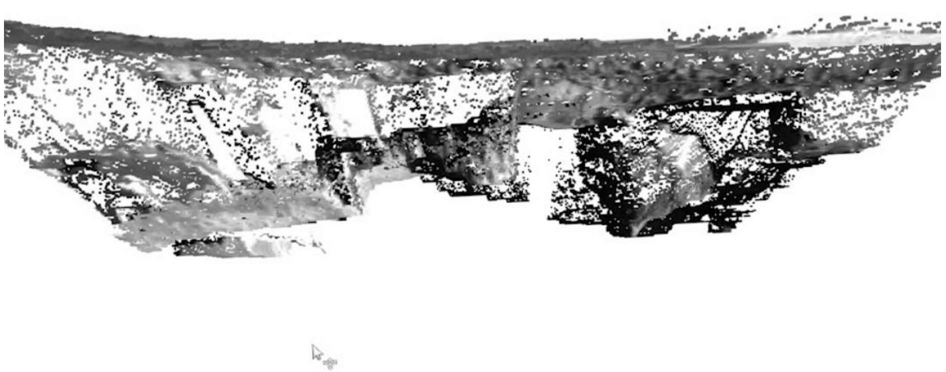


[Pizzoli, Forster, Scaramuzza, REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time ICRA'14]

REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time, ICRA'14, by Pizzoli, Forster, Scaramuzza

- Tracks every pixel (like DTAM) but **Probabilistically**
- Runs live on video streamed from MAV (50 Hz on GPU)
- Copes well with low texture surfaces

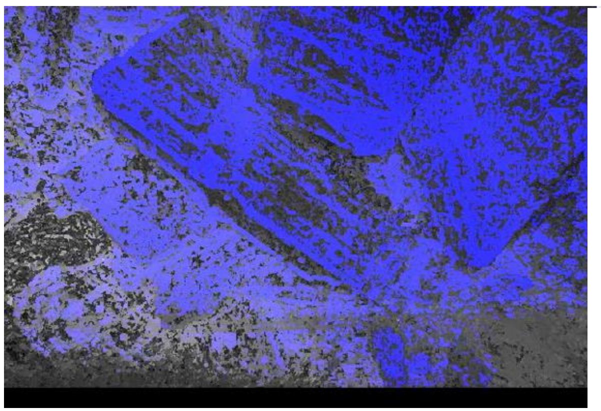


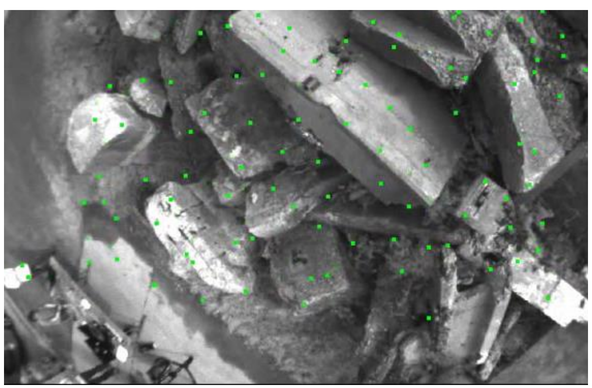


[Pizzoli, Forster, Scaramuzza, REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time ICRA'14]

REMODE applied to autonomous flying 3D scanning

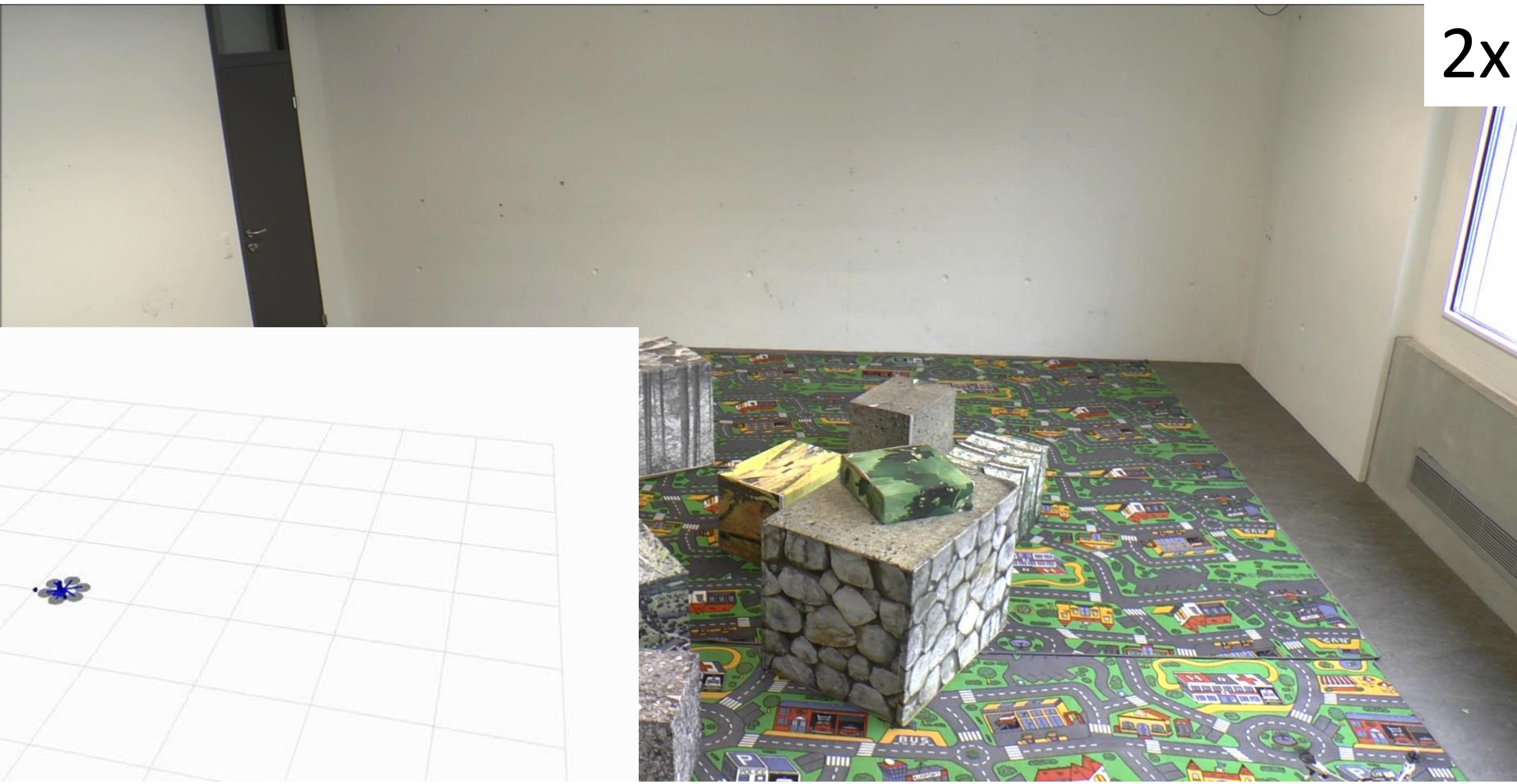
- Tracks every pixel (like DTAM) but **Probabilistically**
- Runs live on video streamed from MAV (50 Hz on GPU)
- Copes well with low texture surfaces





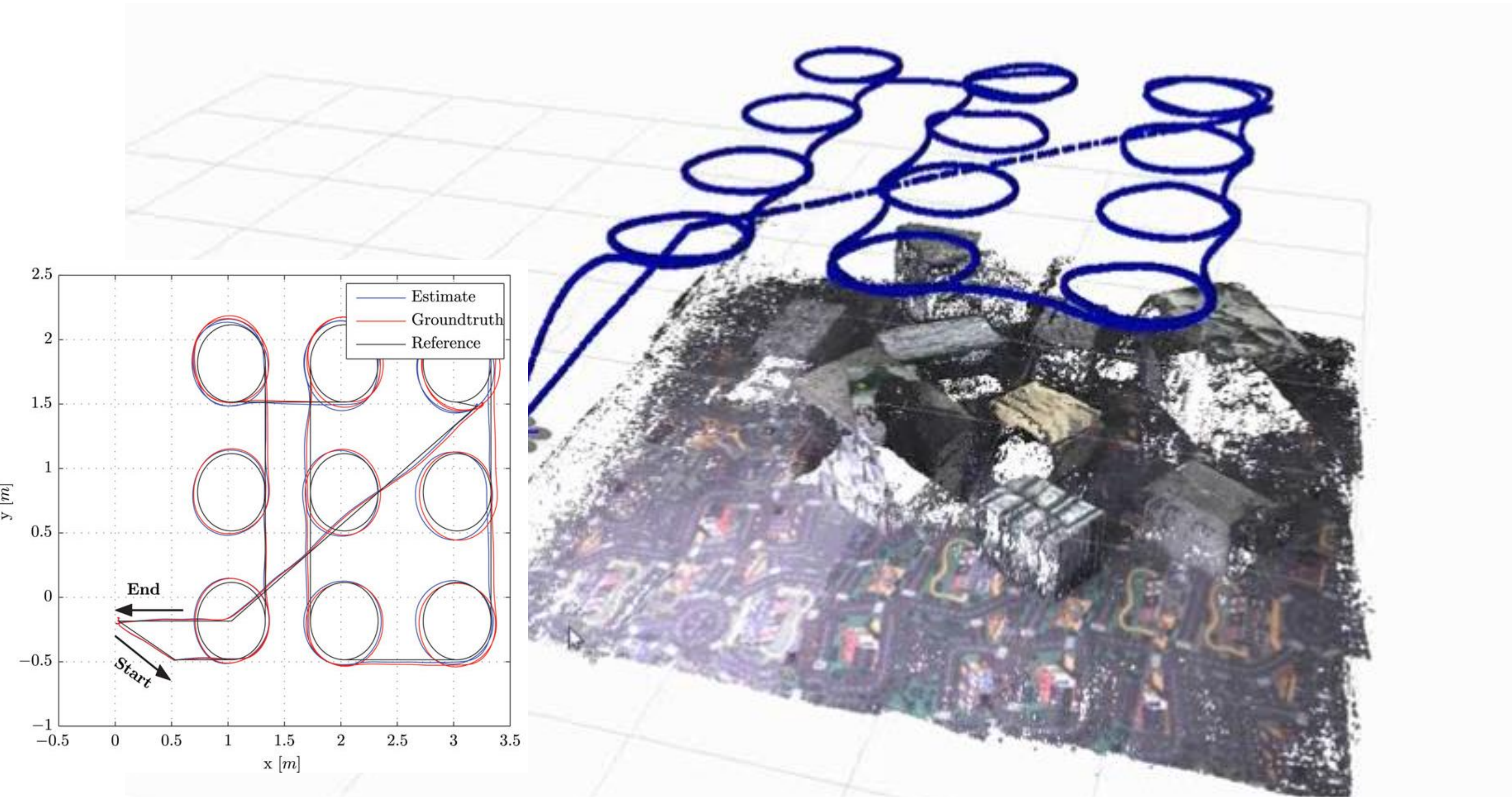
Live demonstration at the Firefighter Training Area of Zurich
Featured on ARTE Tv channel on November 22 and SRF 10vo10

REMODE applied to autonomous flying 3D scanning



Faessler, Fontana, Forster, Mueggler, Pizzoli, Scaramuzza, Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle, **Journal of Field Robotics, 2015**.

REMODE applied to autonomous flying 3D scanning



Faessler, Fontana, Forster, Mueggler, Pizzoli, Scaramuzza, Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle, **Journal of Field Robotics, 2015**.

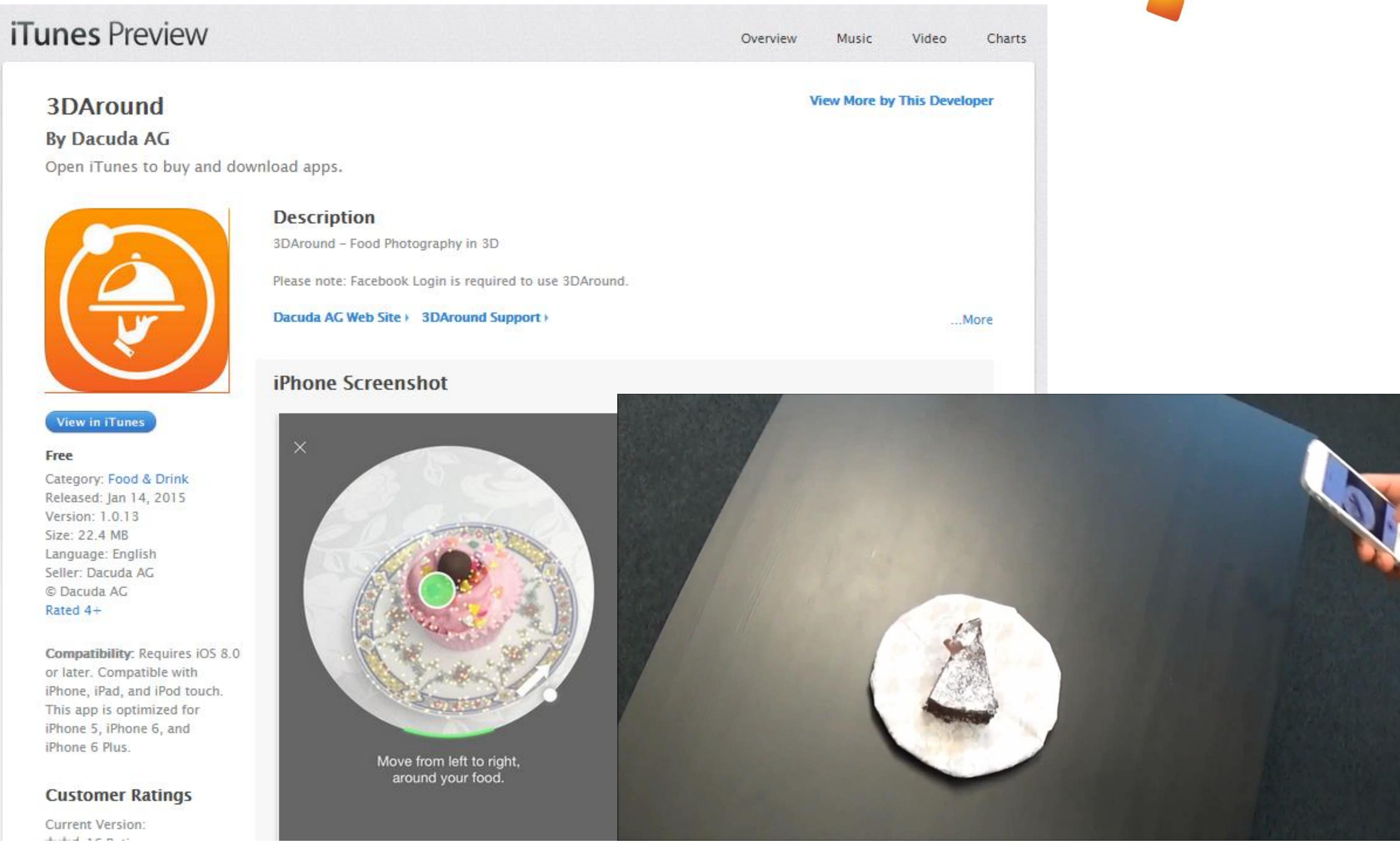
REMODE applied to autonomous flying 3D scanning

Live demonstration at the Firefighter Training Area of Zurich
Featured on ARTE Tv channel on November 22 and SRF 10vo10

[Pizzoli, Forster, Scaramuzza, REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time ICRA'14]

3DAround iPhone App

Dacuda



Things to remember

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

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