Lecture 10
Multi-view Stereo
(3D Dense Reconstruction)

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

Stage-set from Gruber et al., "The City of Sights", ISMAR'10.
3D Reconstruction from Multiple views

Assumption
• Cameras are calibrated
  – both intrinsically
    • $K$ matrix for each camera is known
  – and extrinsically
    • relative positions $T$ and orientations $R$ between cameras are known (for instance, from SFM)
Multi-view stereo

Input: calibrated images from several viewpoints
Output: 3D object model
Review: The Epipolar Plane

The two camera centers and the feature \( p \) determine a plane called the "epipolar plane", which intersect each camera image plane into an epipolar line.
Thanks to the epipolar constraint, corresponding points only need to be searched along epipolar lines.
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

**Step 2: Global methods**
- Refine the depth surface as a whole by enforcing smoothness constraint
Photometric error (SSD or SAD)

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

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

- Dense reconstruction requires establishing dense correspondences
- Correspondences are computed based on photometric error:
  - patch-based correlation (SAD, SSD, NCC)
  - Difference among pixel intensity values (patch 1x1 pixels)
  - What are the pros and cons of using small or large patches?
- 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]
• Photometric error for flat regions or edges parallel to the epipolar line show multiple minima (because of noise, lack of textures or repetitive textures)
• For distinctive pixels (as in \( b \)) the aggregated photometric error has typically one clear minimum.
Disparity Space Image (DSI)

• For discrete depth hypotheses the aggregate photometric error with respect to the reference image can be stored in the Disparity Space Image (DSI)

\[ C(u, v, d) = \sum_k \rho(\tilde{I}_k(u', v', d) - I_r(u, v)) \]

\( \tilde{I}_k(u', v', d) \) is the pixel in the \( k \)-th image associated with the pixel \((u', v')\) in the reference image \( I_r \) and depth hypothesis \( d \)

• \( \rho(\cdot) \) is the photometric error (e.g., SSD, SAD)

[Szeliski and Golland 1999]
Disparity Space Image (DSI)

240 x 180 x 100 voxels
The solution to the depth estimation problem is a function \( d(u, v) \) in the DSI that presents some **optimality properties**: 

- **Minimum aggregated photometric cost**  \( \arg \min_d C \)

  \[ \text{AND (optionally)} \]

- **best piecewise smoothness** (global methods)
Solution

The solution to the depth estimation problem is a function $d(u, v)$ in the DSI that presents some optimality properties:

- Minimum aggregated photometric cost $\arg \min_d C$

AND (optionally)

- best piecewise smoothness (global methods)
Solution

The solution to the depth estimation problem is a function $d(u, v)$ in the DSI that presents some optimality properties:

- Minimum aggregated photometric cost $\arg \min_d C$

AND (optionally)

- best piecewise smoothness (global methods)
Solution

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

$$E(d) = E_d(d) + \lambda E_s(d)$$

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

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

- $\rho_d$ is a norm (e.g. $L_2$, $L_1$ or Huber norm)
- $\lambda$ controls the tradeoff data / regularization. What happens for large $\lambda$?
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 $\lambda$
[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 $\lambda$
[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 $\lambda$
[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 $\lambda$ [Newcombe et al. 2011]
Regularized depth maps

- Popular assumption: *discontinuities in intensity* coincide with *discontinuities in depth*

- **Control smoothness penalties** according to image gradient
  \[
  \rho_d \left( d(u, v) - d(u + 1, v) \right) \cdot \rho_I \left( \| I(u, v) - I(u + 1, v) \| \right)
  \]

- \( \rho_I \) is some monotonically decreasing function of intensity differences: *lowers smoothness costs at 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 new depth computation
Fusion of multiple depth maps

depth map 1

Sensor

↓

depth map 2

Sensor

combination

depth map 1 + depth map 2 = combination
Fusion of multiple depth maps
Depth map fusion

input image

317 images (hemisphere)

ground truth model

Goesele, Curless, Seitz, 2006
GPGPU

- **GPGPU** = General Purpose computing on Graphics Processing Unit
- 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 sequential serial processing
- More transistors devoted to data processing

[ALU: Arithmetic Logic Unit:](https://www.youtube.com/watch?v=P28LKWZrl)

https://www.youtube.com/watch?v=-P28LKWZrl
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
  - Warping (e.g., epipolar rectification, homography)
  - Feature extraction (i.e., convolutions)

- **Multi-view geometry**
  - Search for dense correspondence
    - *Pixel-wise* operations (correlation)
    - Matrix and vector operations (epipolar geometry)
  - Photometric Cost Aggregation

- **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 quad core 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
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
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

REMODE applied to autonomous flying 3D scanning

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 Apple App

By Dacuda AG

Open iTunes to buy and download apps.

Description

3DAround - Food Photography in 3D

Please note: Facebook Login is required to use 3DAround.

Dacuda AG Web Site › 3DAround Support ›

iPhone Screenshot

Move from left to right, around your food.

Customer Ratings

Latest Version: 1.0.13

Size: 22.4 MB

Language: English

Seller: Dacuda AG

Rated 4–

Compatibility: Requires iOS 8.0 or later. Compatible with iPhone, iPad, and iPod touch. This app is optimized for iPhone 5, iPhone 5S, and iPhone 6 Plus.