High-Quality 3D Reconstruction from RGB-D Sensors

Computer Vision I: Variational Methods

Current Research

Robert Maier

robert.maier@in.tum.de
https://vision.in.tum.de/members/maierr

January 25, 2018

Overview

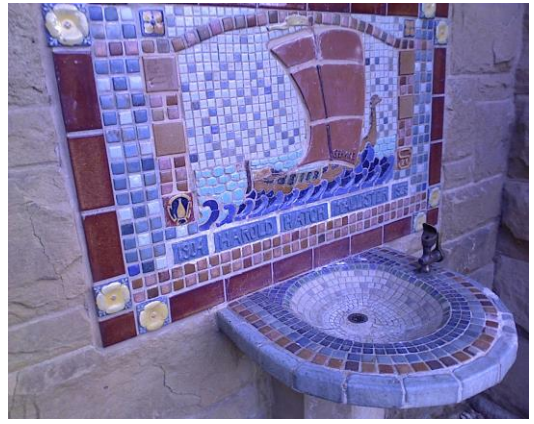
- RGB-D Sensors & 3D Reconstruction
- Signed Distance Functions
- Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions (Bylow et al, RSS 2013)
- De-noising, Stabilizing and Completing 3D Reconstructions On-the-go using Plane Priors (Dzitsiuk et al, ICRA 2017)
- Intrinsic3D: High-Quality 3D Reconstruction by Joint Appearance and Geometry Optimization with Spatially-Varying Lighting (Maier et al, ICCV 2017)

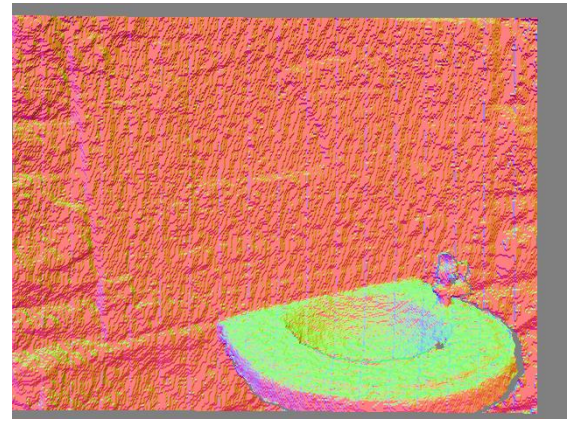
Overview

- RGB-D Sensors & 3D Reconstruction
- Signed Distance Functions
- Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions (Bylow et al, RSS 2013)
- De-noising, Stabilizing and Completing 3D Reconstructions On-the-go using Plane Priors (Dzitsiuk et al, ICRA 2017)
- Intrinsic3D: High-Quality 3D Reconstruction by Joint Appearance and Geometry Optimization with Spatially-Varying Lighting (Maier et al, ICCV 2017)

RGB-D Sensors

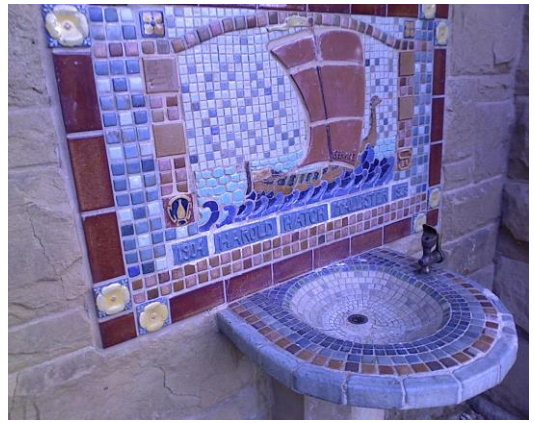
- RGB-D: color (RGB) + depth (metric!) @ 30 fps

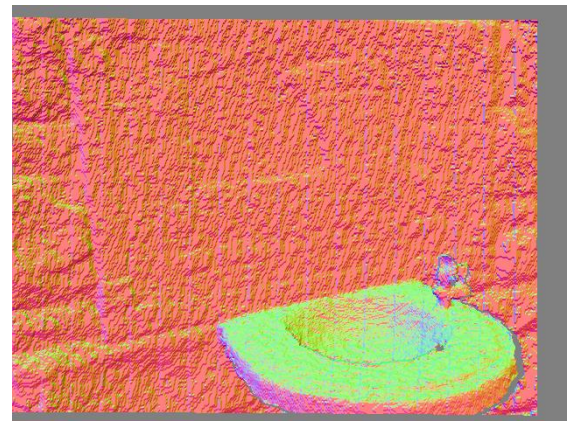




RGB-D Sensors

- RGB-D: color (RGB) + depth (metric!) @ 30 fps
- Structured Light / Time-of-flight
- Low-cost!



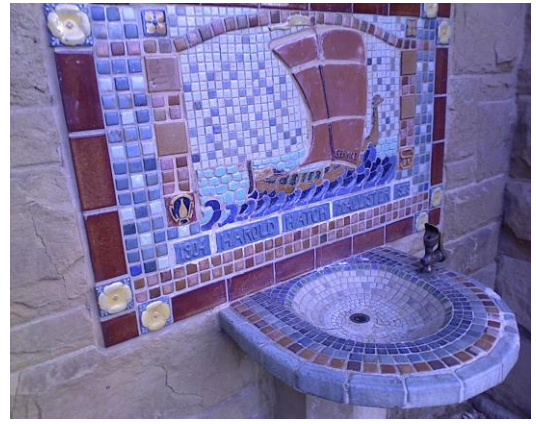


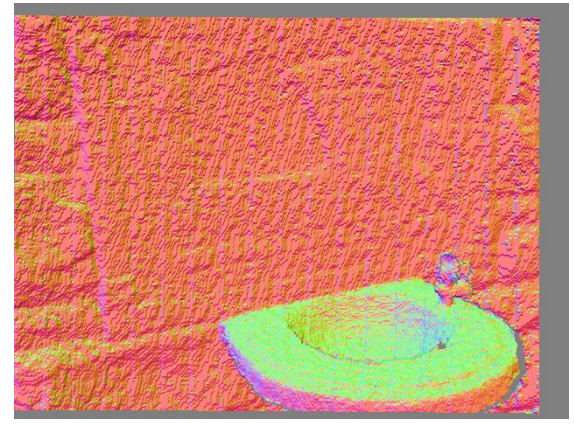
Microsoft Kinect v1

Asus Xtion

RGB-D Sensors

- RGB-D: color (RGB) + depth (metric!) @ 30 fps
- Structured Light / Time-of-flight
- Low-cost!





Microsoft Kinect v1

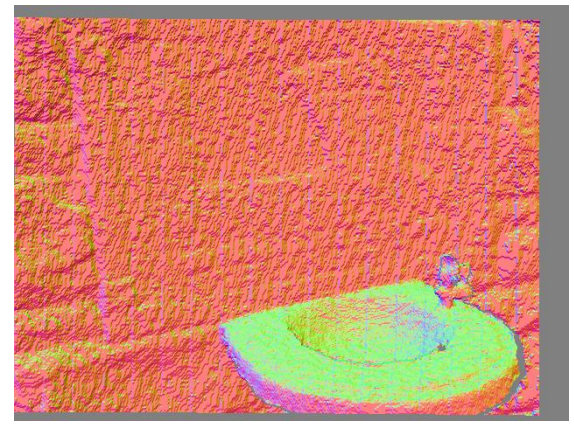
Intel RealSense R200

Asus Xtion

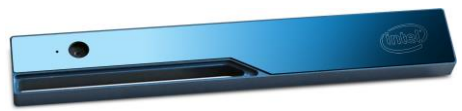
Occipital Structure Sensor

RGB-D Sensors

- RGB-D: color (RGB) + depth (metric!) @ 30 fps
- Structured Light / Time-of-flight
- Low-cost!



Microsoft Kinect v1



Intel RealSense R200

Asus Xtion

Occipital Structure Sensor

Google Tango

Lenovo Phab 2 Pro

Asus ZenFone AR

RGB-D based 3D Reconstruction

- Task: given a stream of **RGB-D frames** of a real-world scene, compute its **3D shape that maximizes the geometric consistency**

RGB-D based 3D Reconstruction

- Task: given a stream of **RGB-D frames** of a real-world scene, compute its **3D shape that maximizes the geometric consistency**
- **SLAM**: Simultaneous Localization and Mapping (RGB-D-SLAM)

RGB-D based 3D Reconstruction

- Task: given a stream of **RGB-D frames** of a real-world scene, compute its **3D shape that maximizes the geometric consistency**
- **SLAM**: Simultaneous Localization and Mapping (RGB-D-SLAM)
- **Fusion** of RGB-D frames in **dense volumetric 3D representation** (focus in this talk: Signed Distance Fields)

KinectFusion [Newcombe et al., ISMAR 2011]

3D Reconstruction Applications

- **Robotics**: real-time 3D reconstruction of large-scale environments (e.g. autonomous drones)

Parrot AR drone

3D Reconstruction Applications

- **Robotics**: real-time 3D reconstruction of large-scale environments (e.g. autonomous drones)
- Requirement of high-quality 3D content for **Augmented Reality**, **Virtual Reality**, …

Parrot AR drone

HTC Vive

Microsoft HoloLens

NVIDIA VR Funhouse

3D Reconstruction Applications

- **Robotics**: real-time 3D reconstruction of large-scale environments (e.g. autonomous drones)
- Requirement of high-quality 3D content for **Augmented Reality**, **Virtual Reality**, …
  - Usually: manual modelling (e.g. Maya)

Parrot AR drone

HTC Vive

Microsoft HoloLens

NVIDIA VR Funhouse

3D Reconstruction Applications

- **Robotics**: real-time 3D reconstruction of large-scale environments (e.g. autonomous drones)
- Requirement of high-quality 3D content for **Augmented Reality**, **Virtual Reality**, …
  - Usually: manual modelling (e.g. Maya)
    - Wide availability of **commodity RGB-D sensors**: efficient methods for 3D reconstruction of real-word scenes

Parrot AR drone

HTC Vive

Microsoft HoloLens

NVIDIA VR Funhouse

3D Reconstruction Applications

- **Robotics**: real-time 3D reconstruction of large-scale environments (e.g. autonomous drones)
- Requirement of high-quality 3D content for **Augmented Reality**, **Virtual Reality**, …
  - Usually: manual modelling (e.g. Maya)
    - Wide availability of **commodity RGB-D sensors**: efficient methods for 3D reconstruction of real-word scenes
- Challenge: **how to reconstruct high-quality 3D models from low-cost depth sensors?**

Parrot AR drone

HTC Vive

Microsoft HoloLens

NVIDIA VR Funhouse

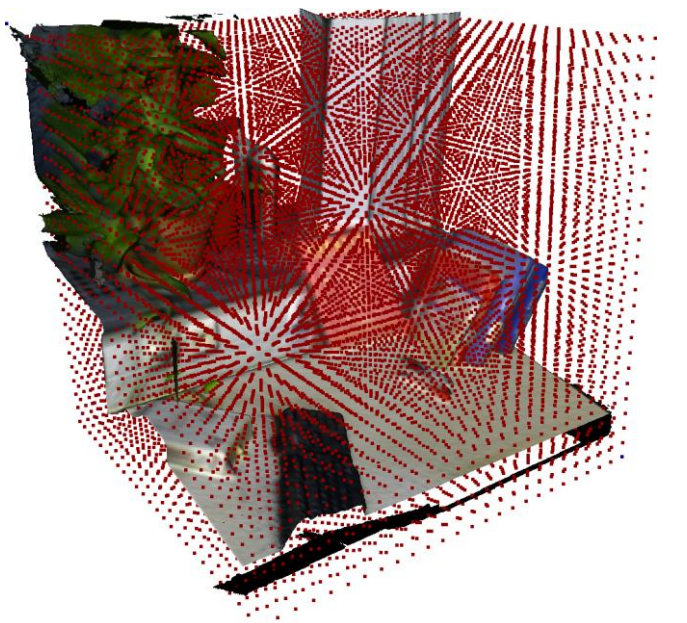
Overview

- RGB-D Sensors & 3D Reconstruction
- Signed Distance Functions
- Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions (Bylow et al, RSS 2013)
- De-noising, Stabilizing and Completing 3D Reconstructions On-the-go using Plane Priors (Dzitsiuk et al, ICRA 2017)
- Intrinsic3D: High-Quality 3D Reconstruction by Joint Appearance and Geometry Optimization with Spatially-Varying Lighting (Maier et al, ICCV 2017)

Signed Distance Functions[1]

Volumetric 3D model representation

- **Voxel grid: dense** (e.g. KinectFusion) or **sparse** (e.g. Voxel Hashing)

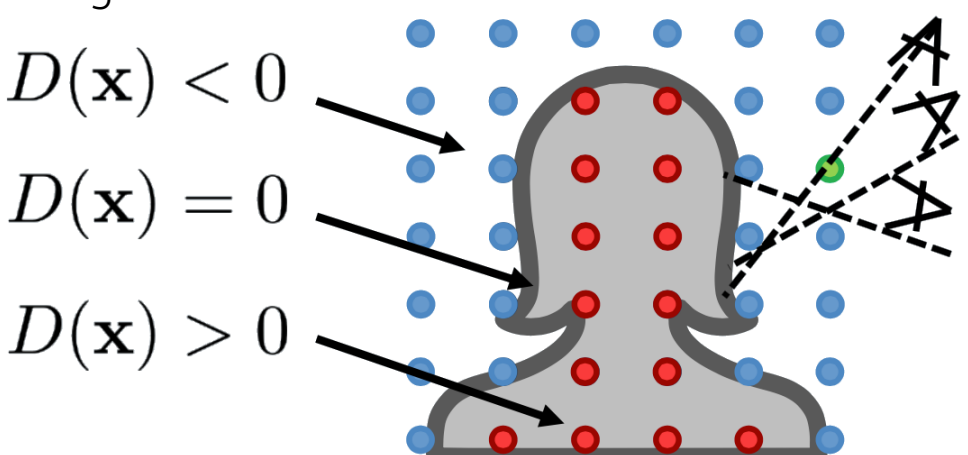


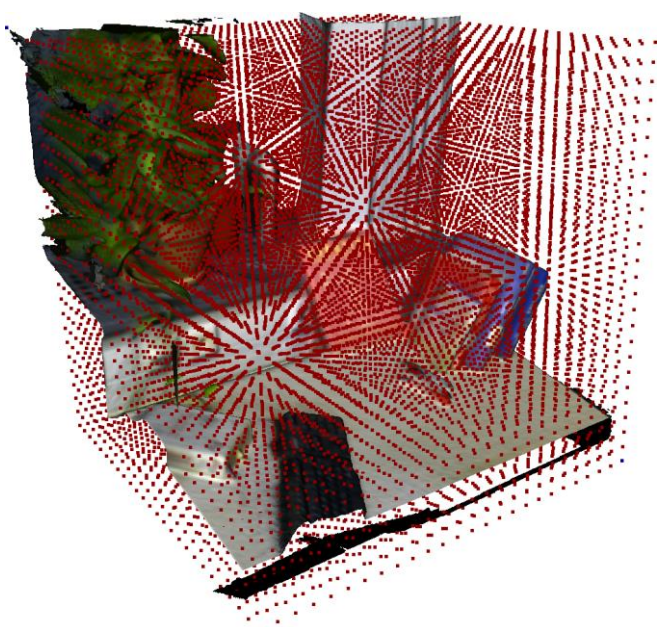
[1] "A volumetric method for building complex models from range images", Curless and Levoy, SIGGRAPH 1996

Signed Distance Functions[1]

Volumetric 3D model representation

- **Voxel grid: dense** (e.g. KinectFusion) or **sparse** (e.g. Voxel Hashing)
- Each voxel stores:
  - Signed Distance Function (SDF): signed distance to closest surface
  - Color values
  - Weights





[1] "A volumetric method for building complex models from range images", Curless and Levoy, SIGGRAPH 1996

Signed Distance Fields

Fusion of depth maps

- Integrate depth maps into SDF with their estimated camera poses

[2] "Marching cubes: A high resolution 3D surface construction algorithm", Lorensen and Cline, SIGGRAPH 1987

Signed Distance Fields

Fusion of depth maps

- Integrate depth maps into SDF with their estimated camera poses
- Voxel updates using weighted average

$$D \leftarrow \frac{WD+wd}{W+w}$$
$$C \leftarrow \frac{WC+wc}{W+w}$$
$$W \leftarrow W + w$$

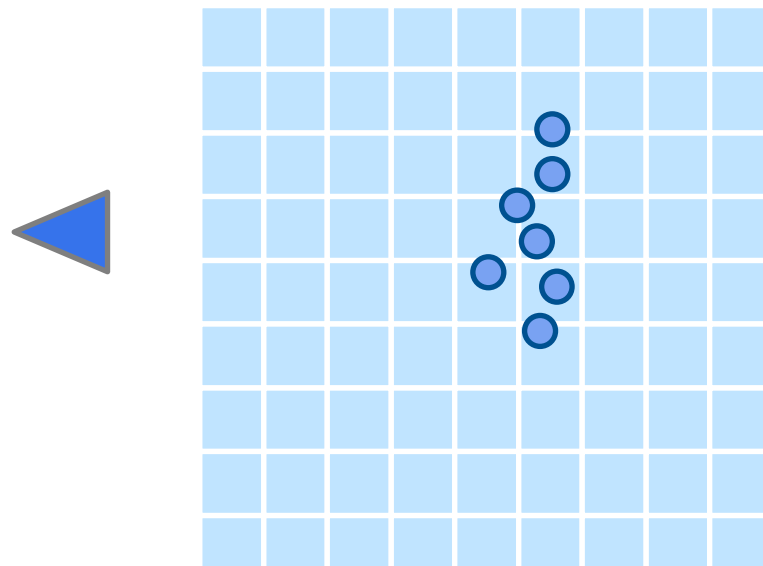
[2] "Marching cubes: A high resolution 3D surface construction algorithm", Lorensen and Cline, SIGGRAPH 1987

Signed Distance Fields

Fusion of depth maps

- Integrate depth maps into SDF with their estimated camera poses
- Voxel updates using weighted average

$$D \leftarrow \frac{WD+wd}{W+w}$$
$$C \leftarrow \frac{WC+wc}{W+w}$$
$$W \leftarrow W + w$$



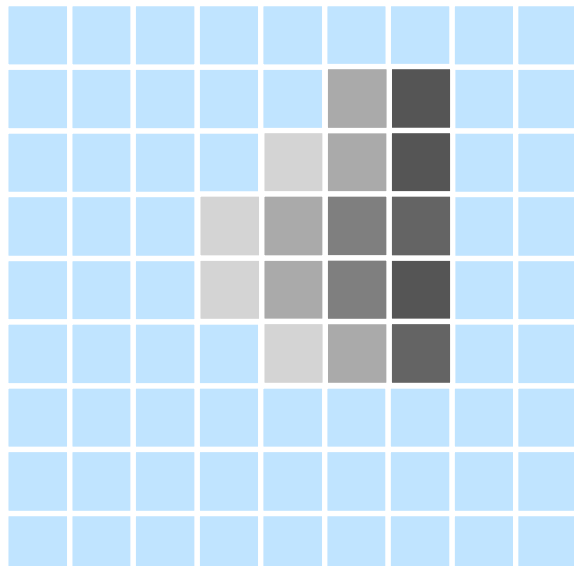
[2] "Marching cubes: A high resolution 3D surface construction algorithm", Lorensen and Cline, SIGGRAPH 1987

Signed Distance Fields

Fusion of depth maps

- Integrate depth maps into SDF with their estimated camera poses
- Voxel updates using weighted average

$$D \leftarrow \frac{WD+wd}{W+w}$$
$$C \leftarrow \frac{WC+wc}{W+w}$$
$$W \leftarrow W + w$$



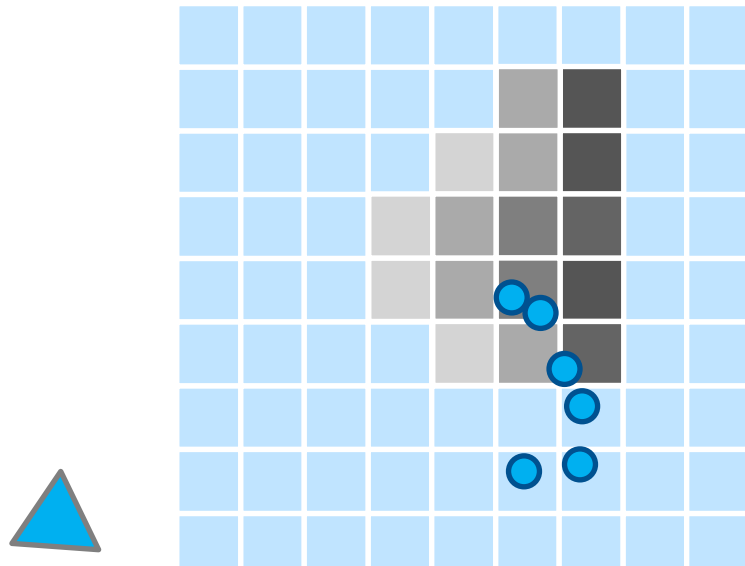
[2] "Marching cubes: A high resolution 3D surface construction algorithm", Lorensen and Cline, SIGGRAPH 1987

Signed Distance Fields

Fusion of depth maps

- Integrate depth maps into SDF with their estimated camera poses
- Voxel updates using weighted average

$$D \leftarrow \frac{WD+wd}{W+w}$$
$$C \leftarrow \frac{WC+wc}{W+w}$$
$$W \leftarrow W + w$$



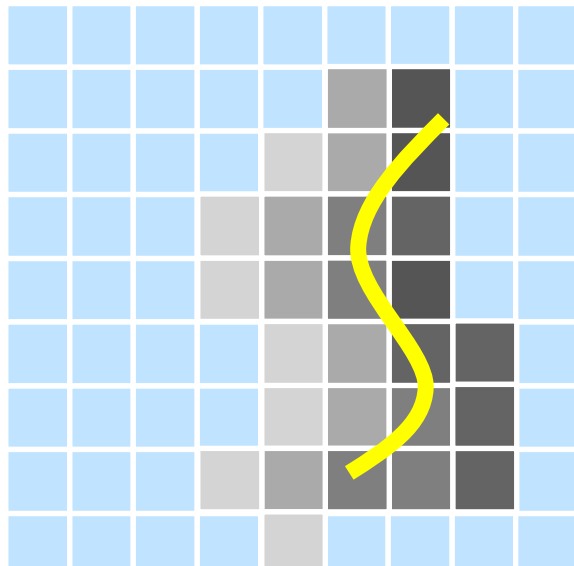
[2] "Marching cubes: A high resolution 3D surface construction algorithm", Lorensen and Cline, SIGGRAPH 1987

Signed Distance Fields

Fusion of depth maps

- Integrate depth maps into SDF with their estimated camera poses
- Voxel updates using weighted average

$$D \leftarrow \frac{WD+wd}{W+w}$$
$$C \leftarrow \frac{WC+wc}{W+w}$$
$$W \leftarrow W + w$$



- Extract ISO-surface with Marching Cubes[2] (triangle mesh)

[2] "Marching cubes: A high resolution 3D surface construction algorithm", Lorensen and Cline, SIGGRAPH 1987

Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions

E. Bylow[1], J. Sturm[2], C. Kerl[2], F. Kahl[1], D. Cremers[2]

[1] Lund University

[2] Technical University of Munich

Robotics: Science and Systems (RSS)
2013, Berlin, Germany

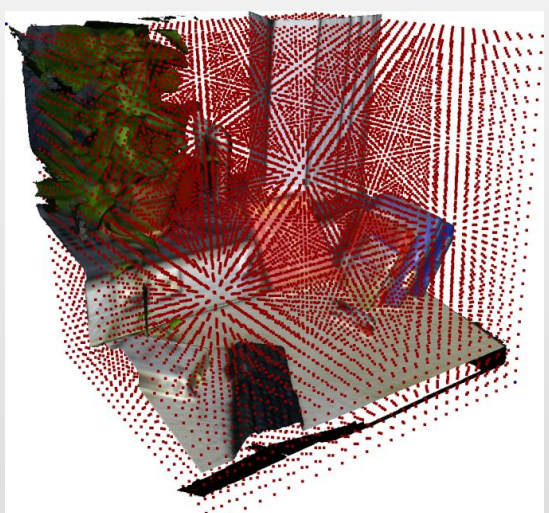
System Pipeline

- Based on KinectFusion (Newcombe et al, ISMAR 2011)

System Pipeline

- Based on KinectFusion (Newcombe et al, ISMAR 2011)

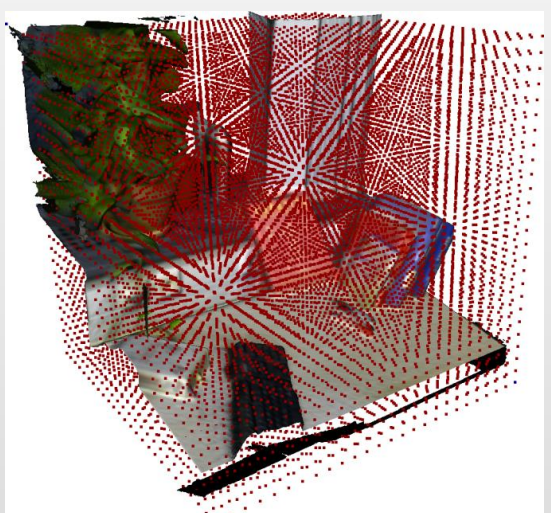
3D model: Signed Distance Field (SDF volume)



System Pipeline

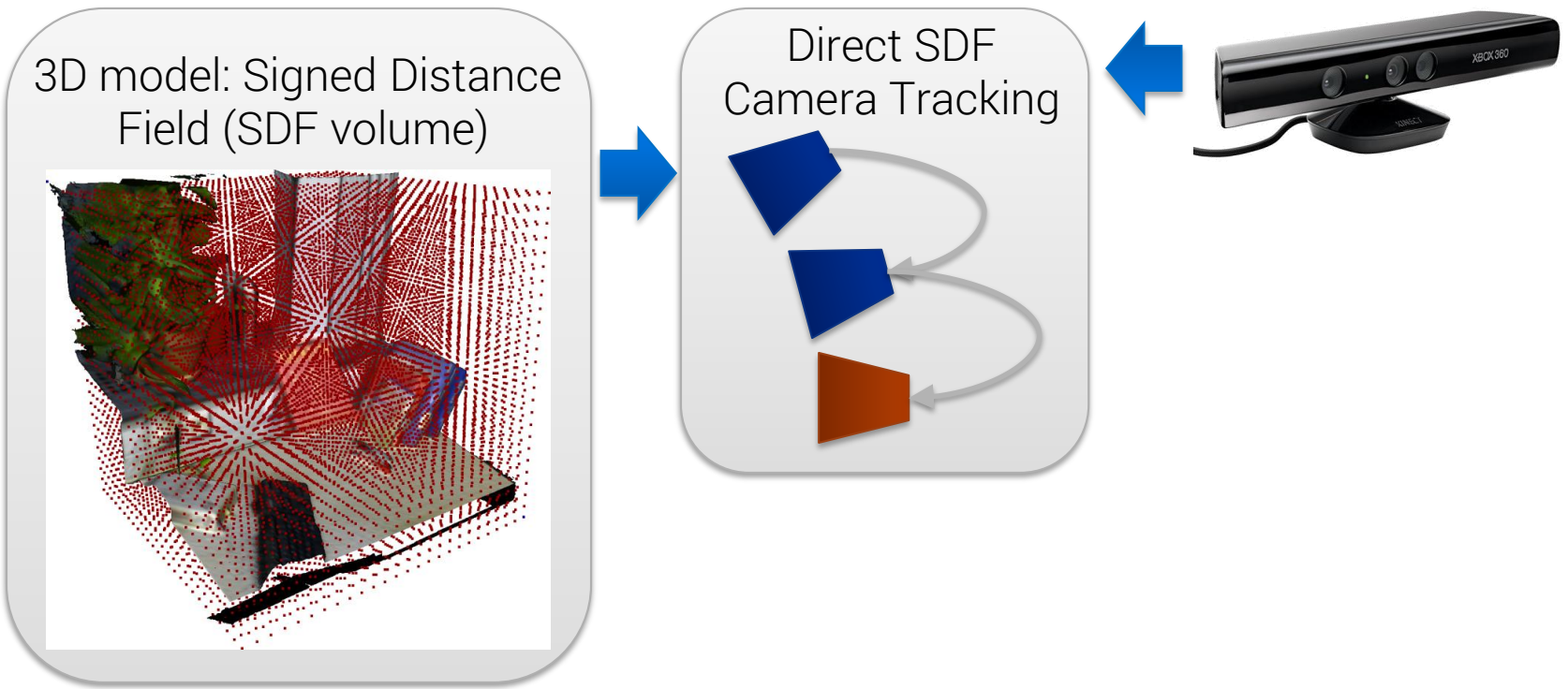
- Based on KinectFusion (Newcombe et al, ISMAR 2011)

3D model: Signed Distance Field (SDF volume)



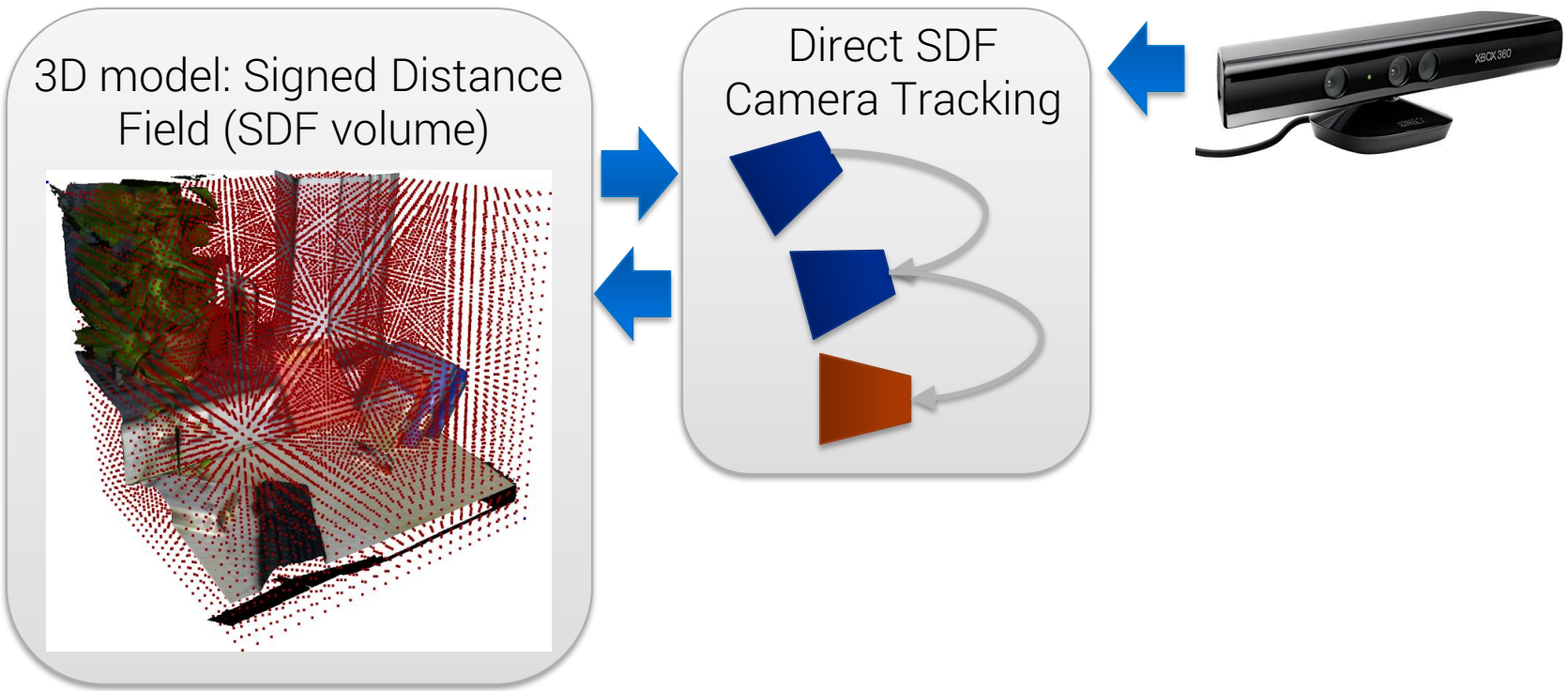
System Pipeline

- Based on KinectFusion (Newcombe et al, ISMAR 2011)



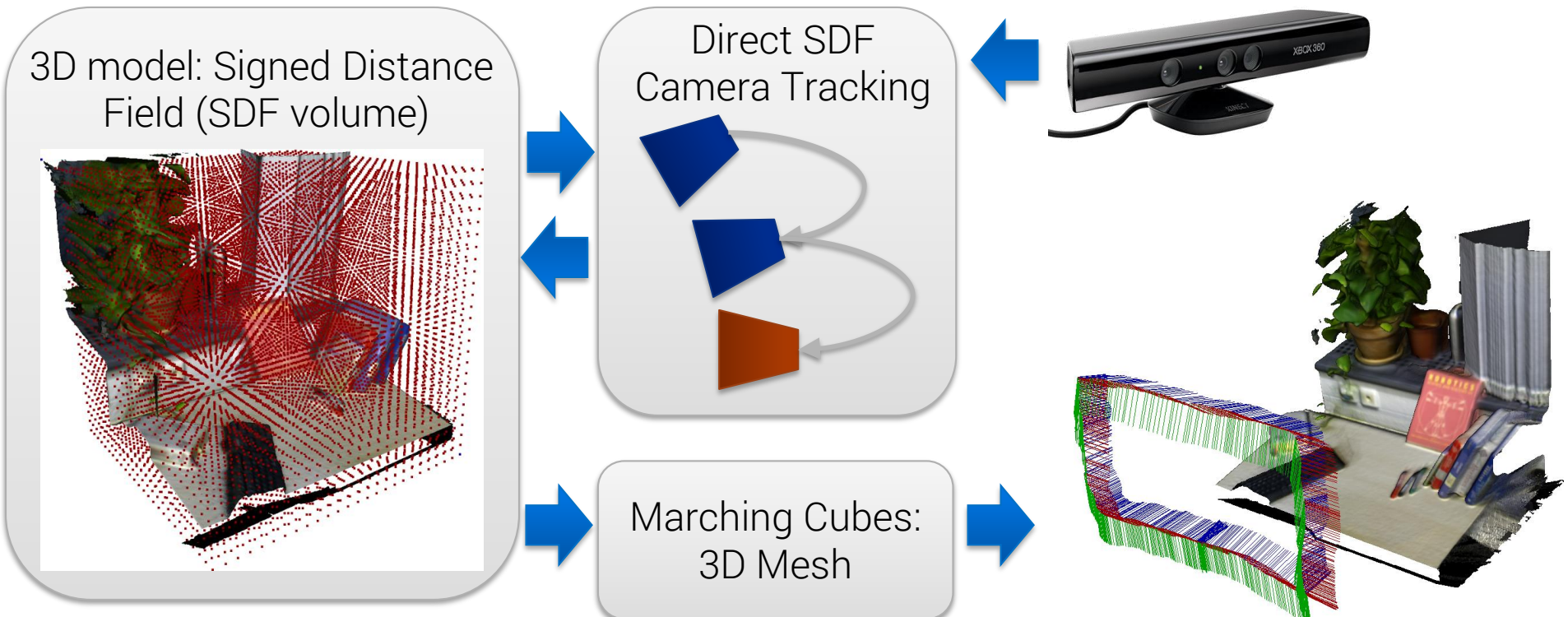
System Pipeline

- Based on KinectFusion (Newcombe et al, ISMAR 2011)



System Pipeline

- Based on KinectFusion (Newcombe et al, ISMAR 2011)



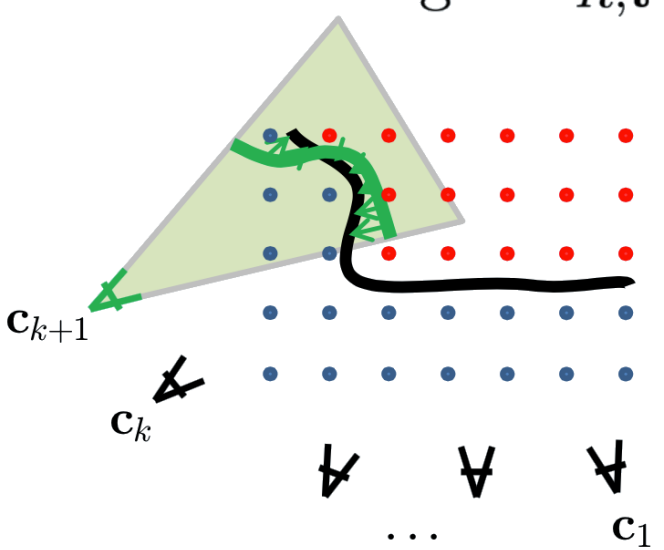
Camera tracking

- **Estimate current camera pose** from input RGB-D frame
- KinectFusion: synthetic depth map from SDF (raycasting) + ICP alignment

Camera tracking

- **Estimate current camera pose** from input RGB-D frame
- KinectFusion: synthetic depth map from SDF (raycasting) + ICP alignment
- Novel direct camera tracking against SDF: **direct minimization of error** between input depth map and SDF:

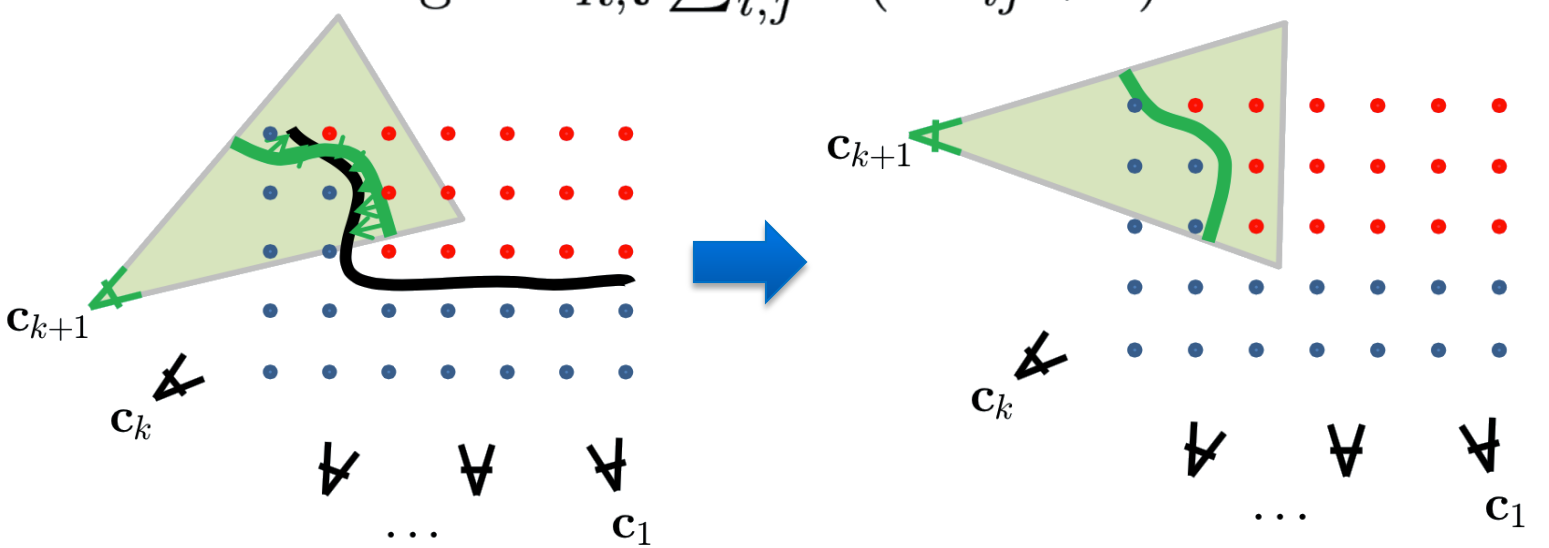
$$\arg\min_{R,\mathbf{t}} \sum_{i,j} D(R\mathbf{x}_{ij} + \mathbf{t})^2$$



Camera tracking

- **Estimate current camera pose** from input RGB-D frame
- KinectFusion: synthetic depth map from SDF (raycasting) + ICP alignment
- Novel direct camera tracking against SDF: **direct minimization of error** between input depth map and SDF:

$$\arg\min_{R,\mathbf{t}} \sum_{i,j} D(R\mathbf{x}_{ij} + \mathbf{t})^2$$



Extension: CopyMe3D

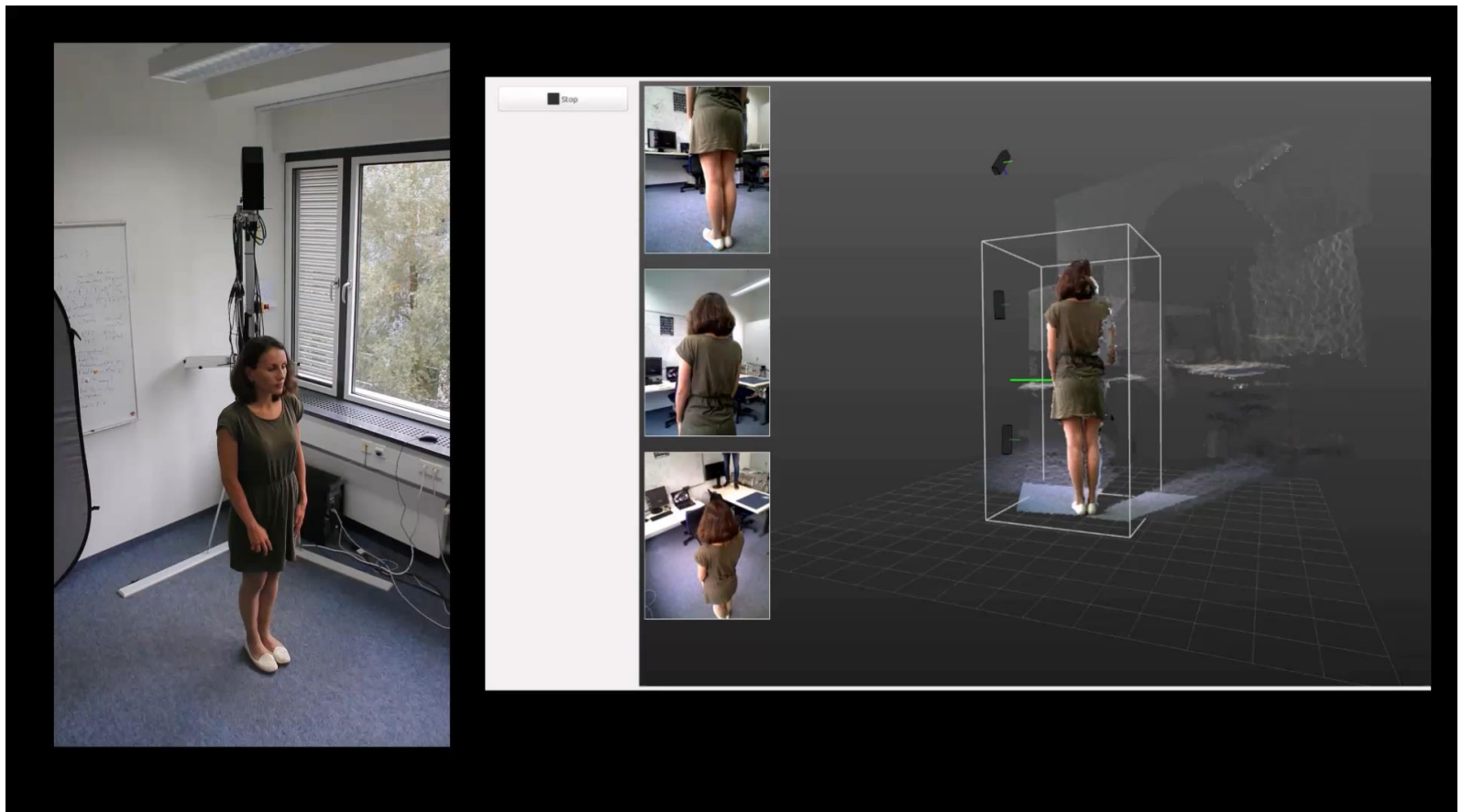
CopyMe3D: Scanning and Printing Persons in 3D (Sturm et al, GCPR 2013)

CopyMe3D

Printed 3D figures

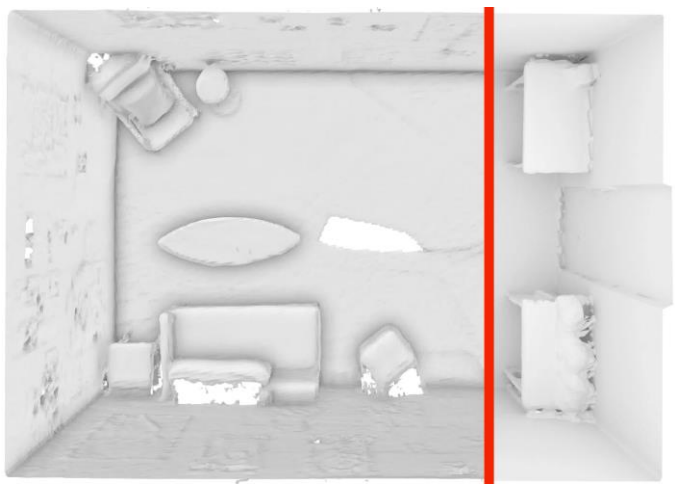
CopyMe3D2

Full-body scanning



De-noising, Stabilizing and Completing 3D Reconstructions On-the-go using Plane Priors

M. Dzitsiuk[1,2], J. Sturm[2], R. Maier[1], L. Ma[1], D. Cremers[1]



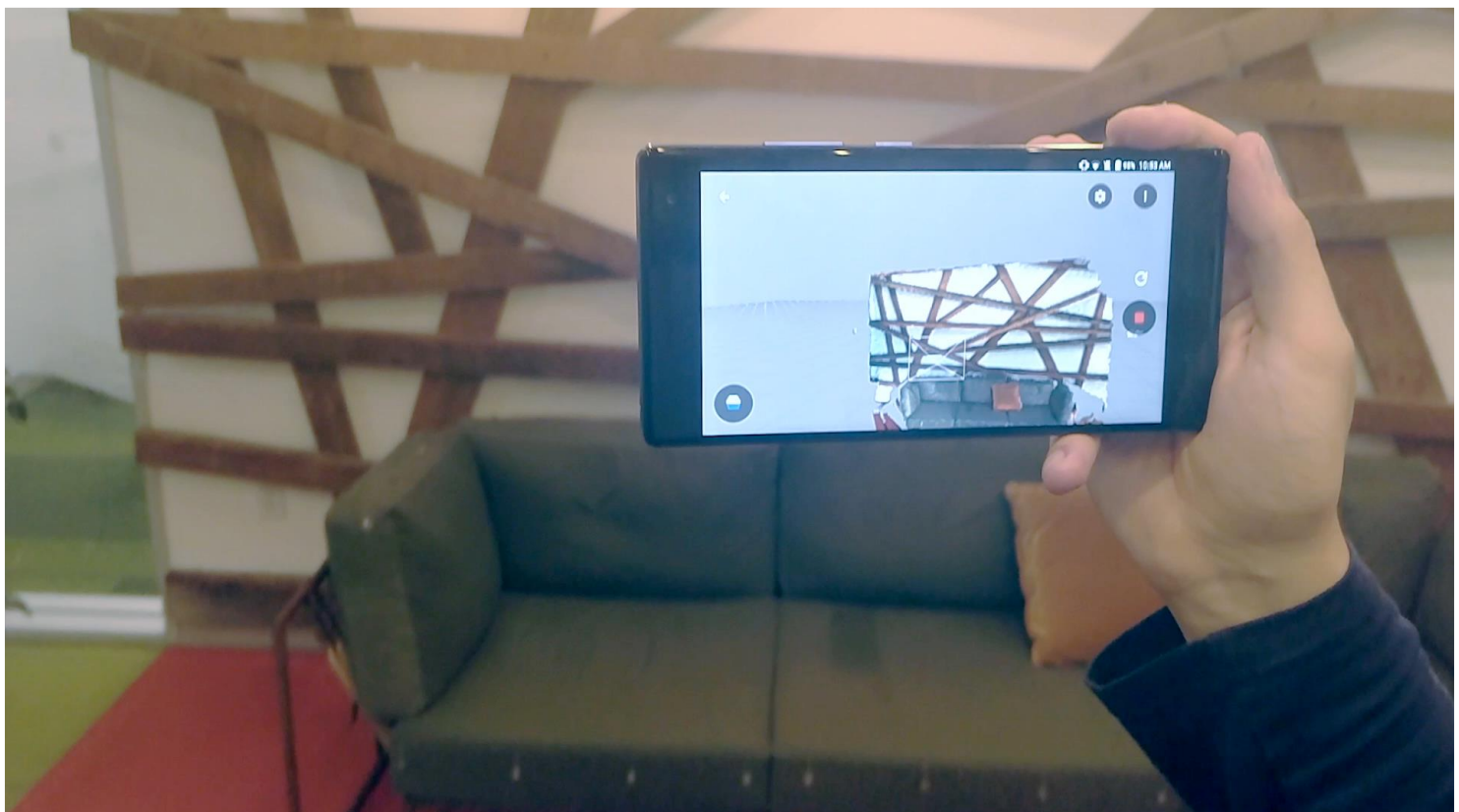
[1] Google

[2] Technical University of Munich

International Conference on Robotics and Automation (ICRA), May 2017, Singapore

3D Reconstruction with Plane Priors

Real-time 3D reconstruction on a mobile device

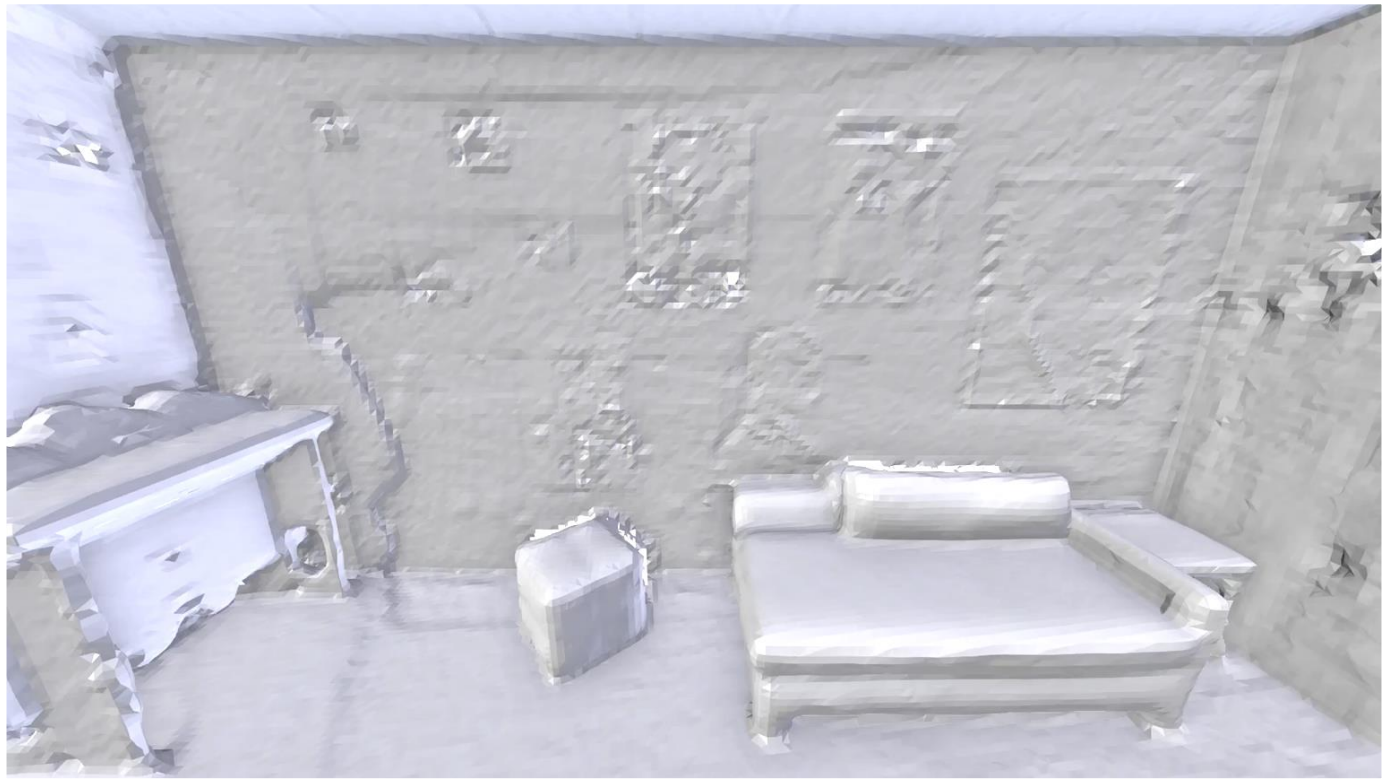


Chisel: Real Time Large Scale 3D Reconstruction Onboard a Mobile Device using Spatially Hashed Signed Distance Fields [Klingensmith et al., 2015]

Tango

3D Reconstruction with Plane Priors

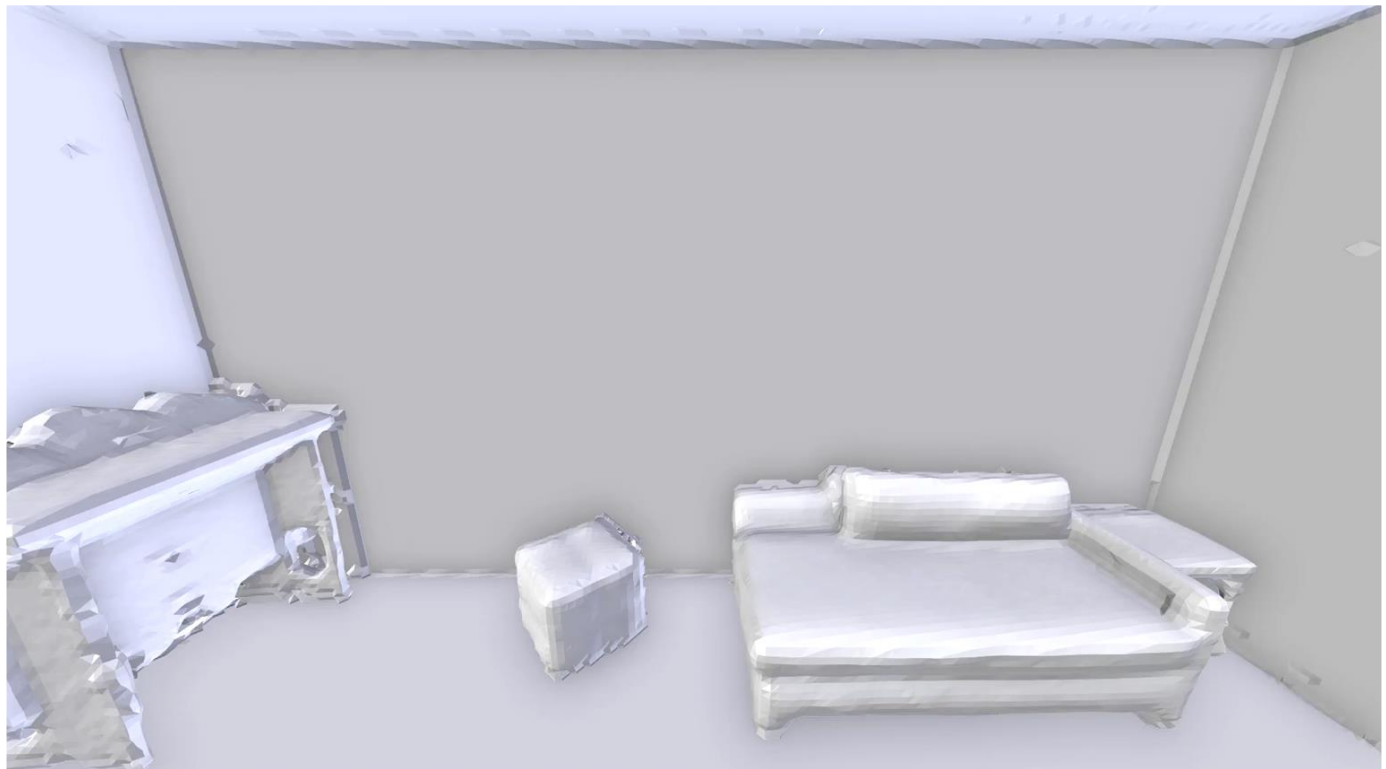
Motivation



Problems with real-time 3D reconstruction: noisy, incomplete, no segmentation

3D Reconstruction with Plane Priors

Motivation

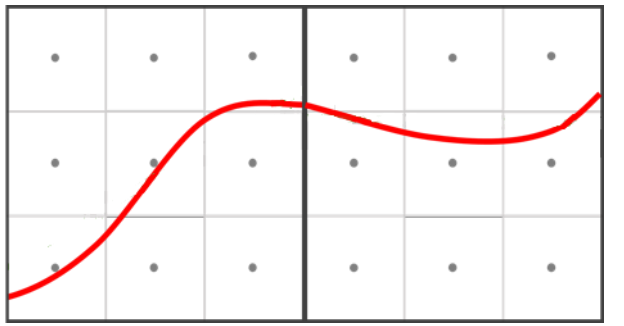


Solution idea: detect and use planes

3D Reconstruction with Plane Priors

Approach

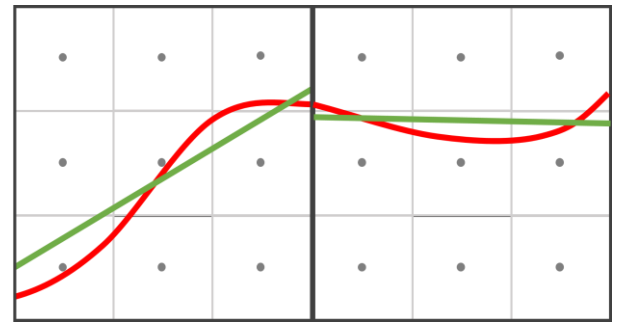
- **Input:** Signed Distance Field divided into voxel blocks



3D Reconstruction with Plane Priors

Approach

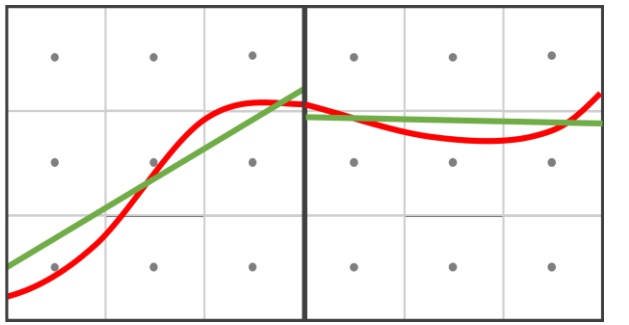
- **Input:** Signed Distance Field divided into voxel blocks
- 1. **Find local plane candidates**



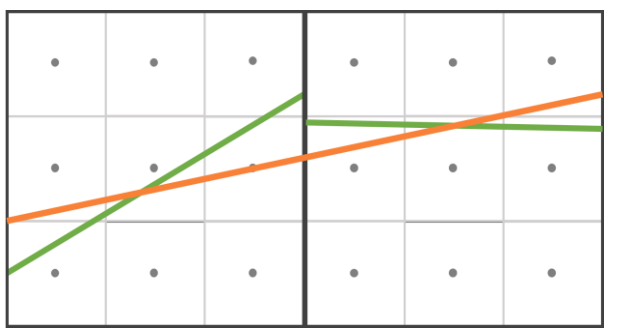
3D Reconstruction with Plane Priors

Approach

- **Input:** Signed Distance Field divided into voxel blocks
- 1. **Find local plane candidates**



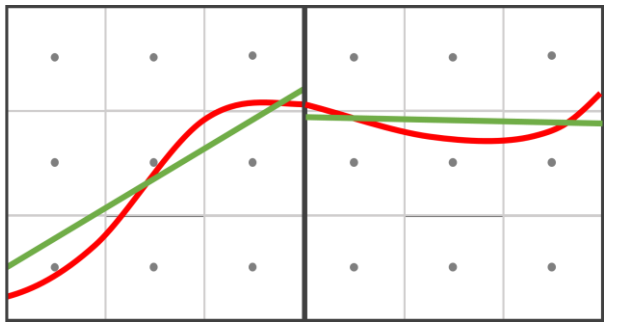
- 2. **Merge planes**



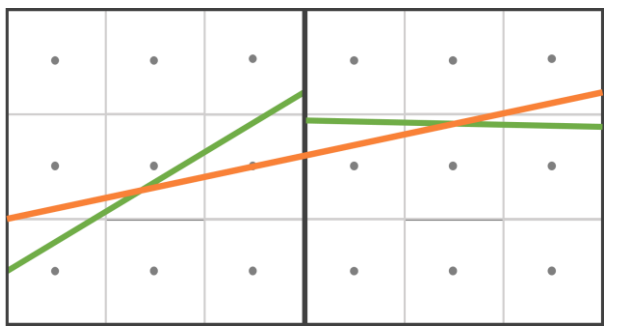
3D Reconstruction with Plane Priors

Approach

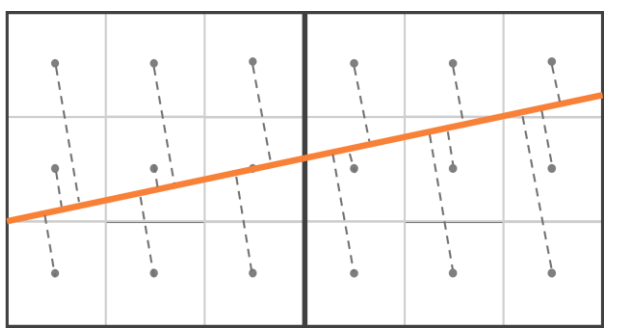
- **Input:** Signed Distance Field divided into voxel blocks
- 1. **Find local plane candidates**



- 2. **Merge planes**



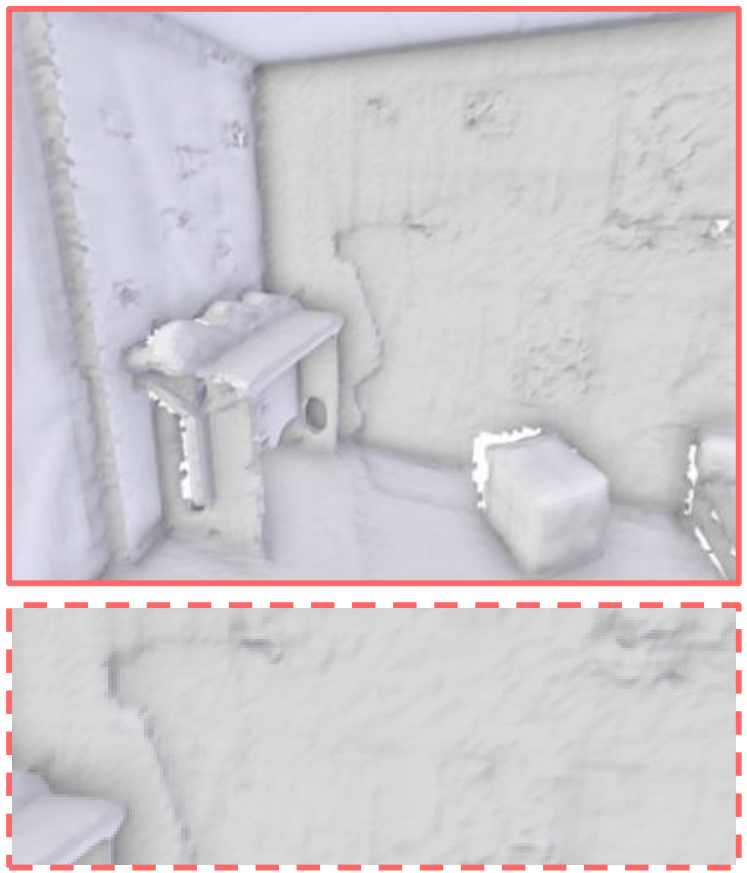
- 3. **De-noising**



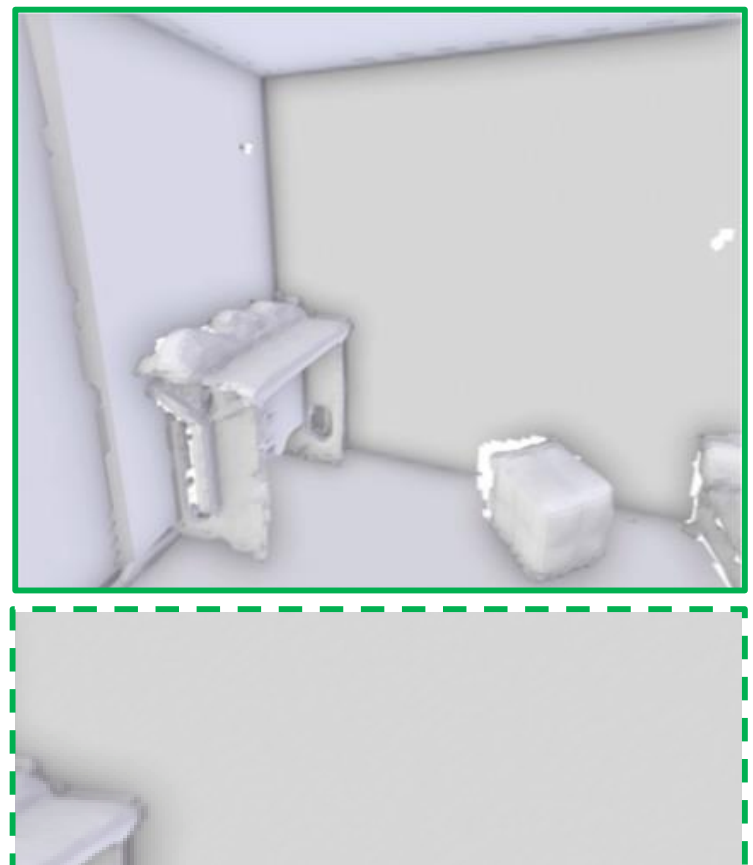
3D Reconstruction with Plane Priors

Results: de-noising

Before



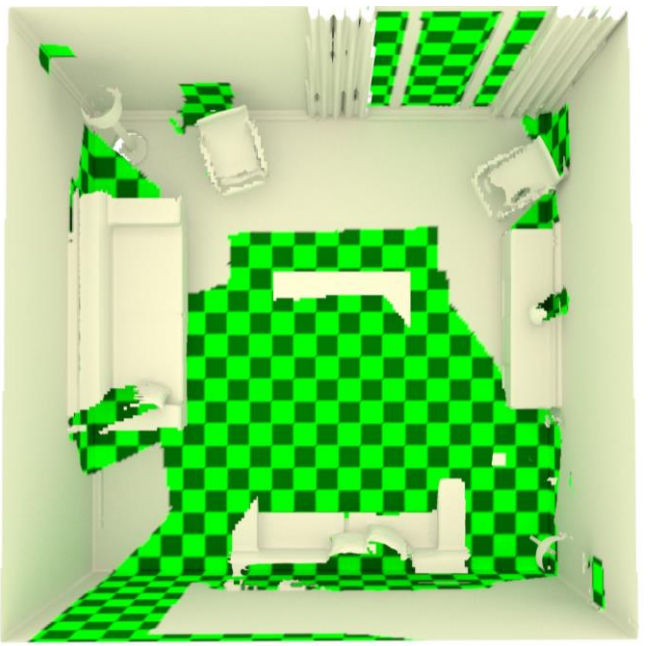
After



3D Reconstruction with Plane Priors

Results

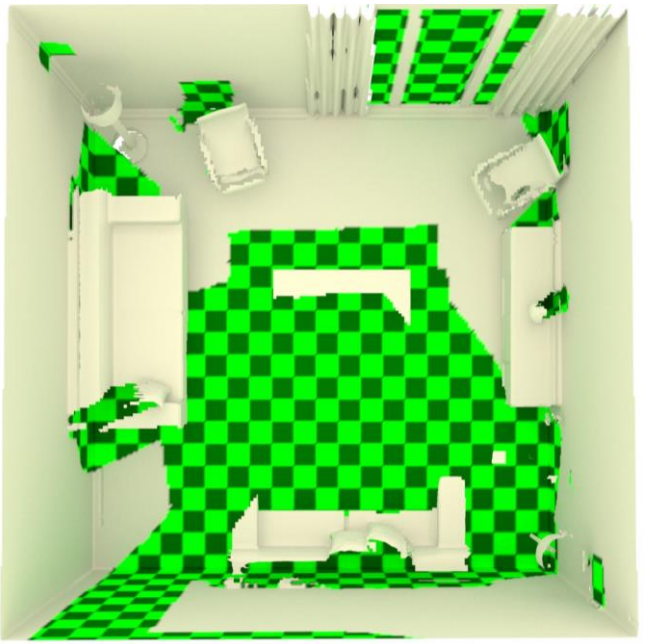
Hole filling



3D Reconstruction with Plane Priors

Results

Hole filling



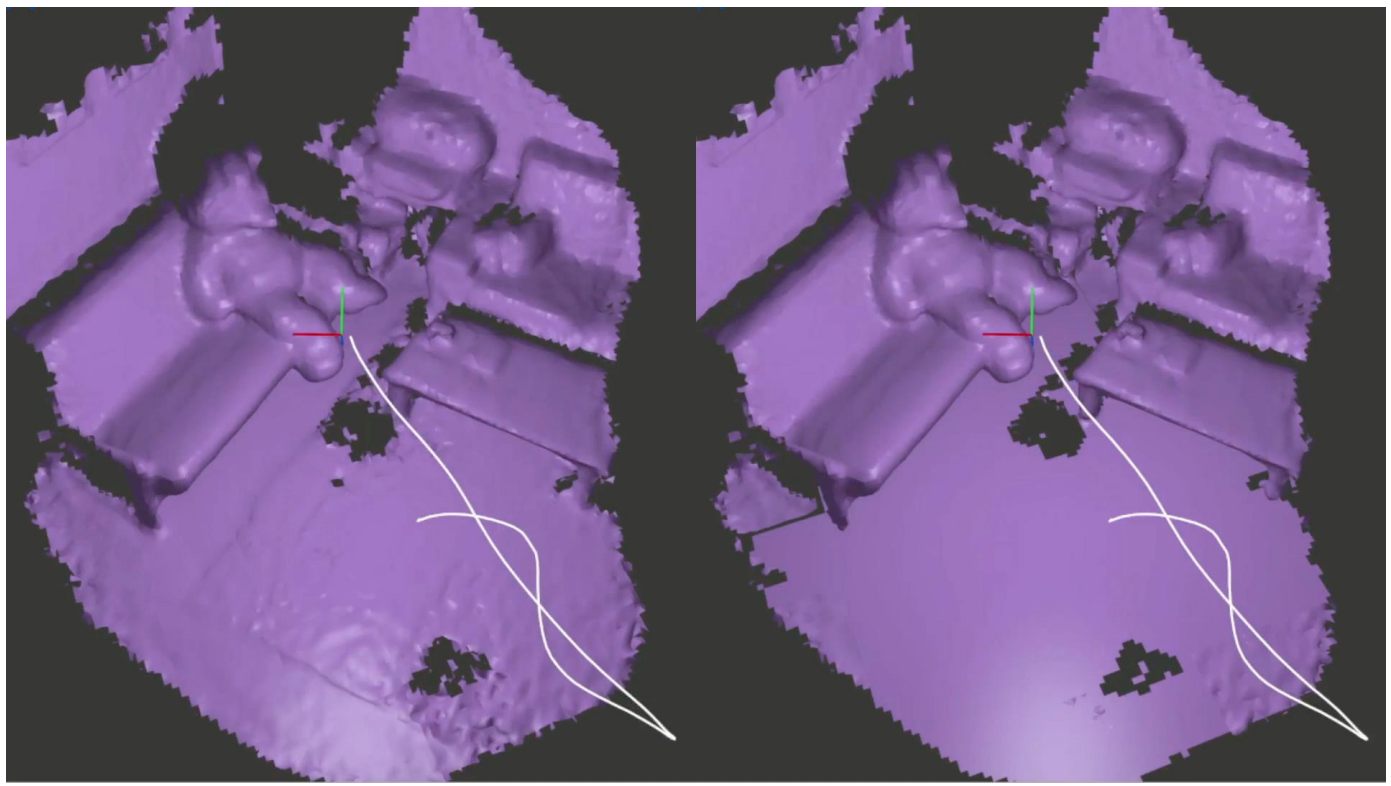
Segmentation

Classify reconstruction geometry:

- **Floor** or **wall** (area, angle with gravity)
- **Object** (mesh connected components)

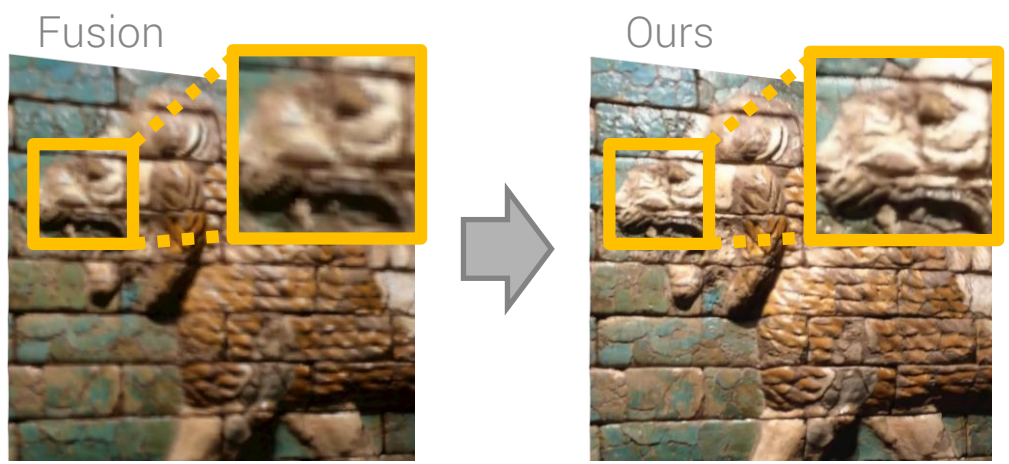
3D Reconstruction with Plane Priors

Results: real-time 3D reconstruction on mobile device



Intrinsic3D: High-Quality 3D Reconstruction by Joint Appearance and Geometry Optimization with Spatially-Varying Lighting

R. Maier[1,2], K. Kim[1], D. Cremers[2], J. Kautz[1], M. Nießner[2,3]



International Conference on Computer Vision (ICCV)
October 2017, Venice, Italy

[1] NVIDIA
[2] TU Munich
[3] Stanford University

State-of-the-art

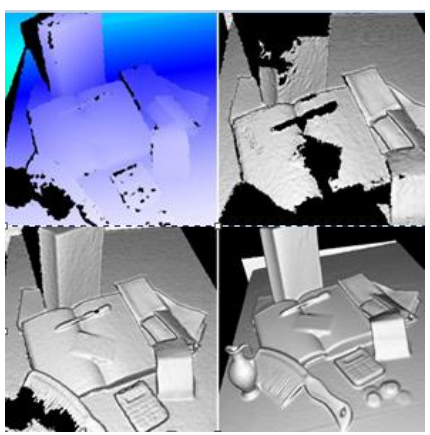
RGB-D based 3D Reconstruction

- Challenge: how to **reconstruct high-quality 3D models** with **best-possible geometry and color** from **low-cost depth sensors?**

State-of-the-art

RGB-D based 3D Reconstruction

- Challenge: how to **reconstruct high-quality 3D models** with **best-possible geometry and color** from **low-cost depth sensors?**
- Real-time, robust, fairly accurate geometric reconstructions



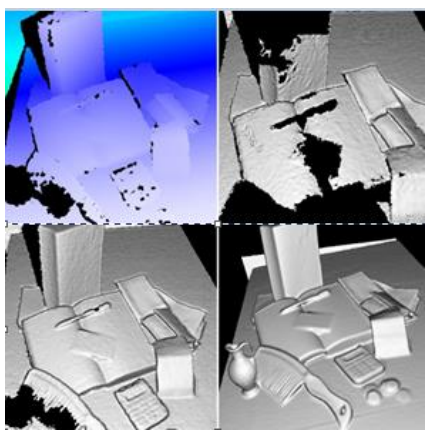
KinectFusion, 2011

"KinectFusion: Real-time Dense Surface Mapping and Tracking", Newcombe et al., ISMAR 2011.

State-of-the-art

RGB-D based 3D Reconstruction

- Challenge: how to **reconstruct high-quality 3D models** with **best-possible geometry and color** from **low-cost depth sensors?**
- Real-time, robust, fairly accurate geometric reconstructions



KinectFusion, 2011

DynamicFusion, 2015

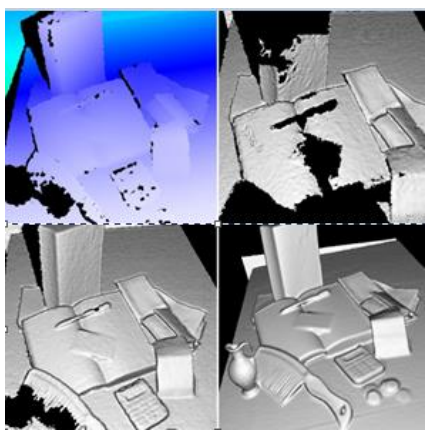
"KinectFusion: Real-time Dense Surface Mapping and Tracking", Newcombe et al., ISMAR 2011.

"DynamicFusion: Reconstruction and Tracking of Non-rigid Scenes in Real-time", Newcombe et al., CVPR 2015.

State-of-the-art

RGB-D based 3D Reconstruction

- Challenge: how to **reconstruct high-quality 3D models** with **best-possible geometry and color** from **low-cost depth sensors?**
- Real-time, robust, fairly accurate geometric reconstructions



KinectFusion, 2011

"KinectFusion: Real-time Dense Surface Mapping and Tracking", Newcombe et al., ISMAR 2011.

DynamicFusion, 2015

"DynamicFusion: Reconstruction and Tracking of Non-rigid Scenes in Real-time", Newcombe et al., CVPR 2015.

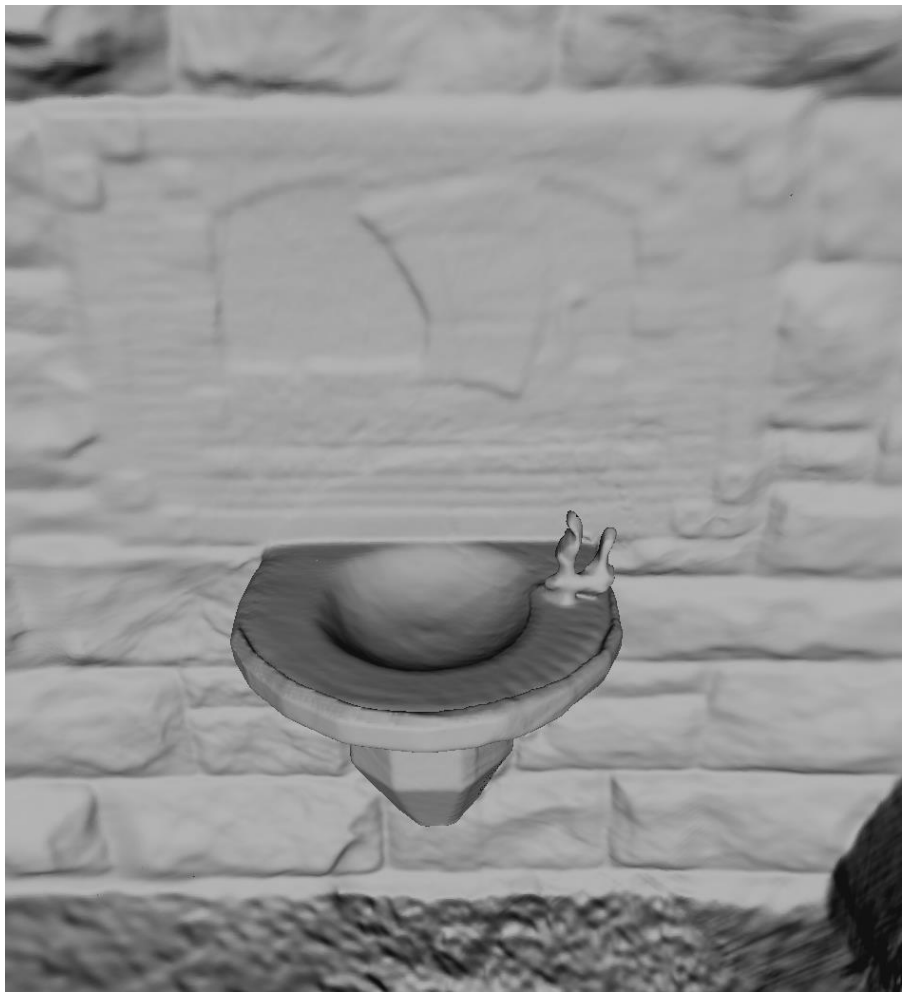
BundleFusion, 2017

"BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration", Dai et al., ToG 2017.

State-of-the-art

Voxel Hashing

- **Baseline** RGB-D based **3D reconstruction framework**
  - initial camera poses
  - sparse SDF reconstruction

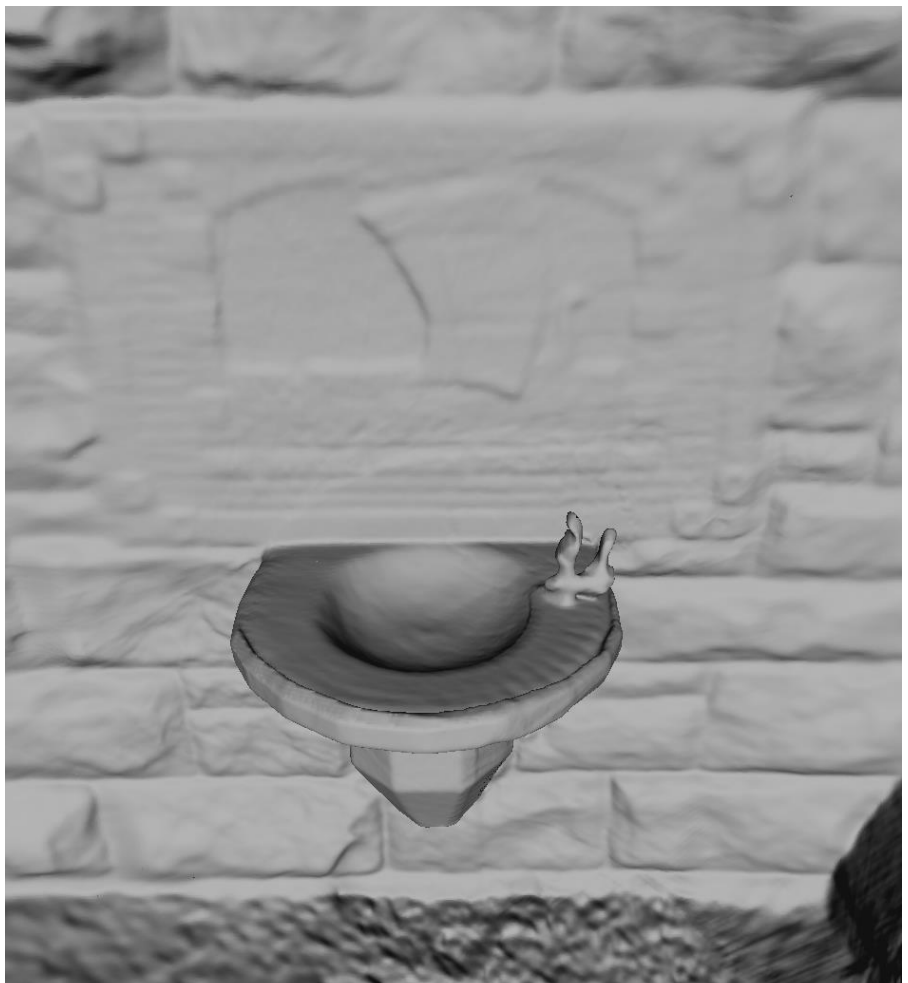


"Real-time 3D Reconstruction at Scale using Voxel Hashing", Nießner et al., ToG 2013.

State-of-the-art

Voxel Hashing

- **Baseline** RGB-D based **3D reconstruction framework**
  - initial camera poses
  - sparse SDF reconstruction
- Challenges:
  - (Slightly) **inaccurate and over-smoothed geometry**



"Real-time 3D Reconstruction at Scale using Voxel Hashing", Nießner et al., ToG 2013.

State-of-the-art

Voxel Hashing

- **Baseline** RGB-D based **3D reconstruction framework**
  - initial camera poses
  - sparse SDF reconstruction
- Challenges:
  - (Slightly) **inaccurate and over-smoothed geometry**
  - **Bad colors**

"Real-time 3D Reconstruction at Scale using Voxel Hashing", Nießner et al., ToG 2013.

State-of-the-art

Voxel Hashing

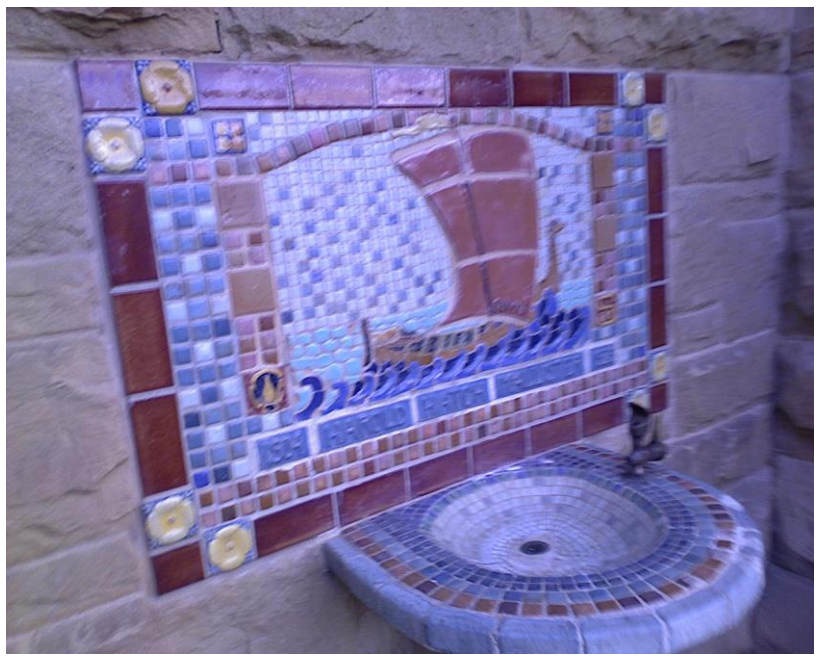
- **Baseline** RGB-D based **3D reconstruction framework**
  - initial camera poses
  - sparse SDF reconstruction
- Challenges:
  - (Slightly) **inaccurate and over-smoothed geometry**
  - **Bad colors**
  - **Inaccurate camera pose** estimation

"Real-time 3D Reconstruction at Scale using Voxel Hashing", Nießner et al., ToG 2013.

State-of-the-art

Voxel Hashing

- **Baseline** RGB-D based **3D reconstruction framework**
  - initial camera poses
  - sparse SDF reconstruction
- Challenges:
  - (Slightly) **inaccurate and over-smoothed geometry**
  - **Bad colors**
  - **Inaccurate camera pose** estimation
  - **Input data** quality (e.g. motion blur, sensor noise)

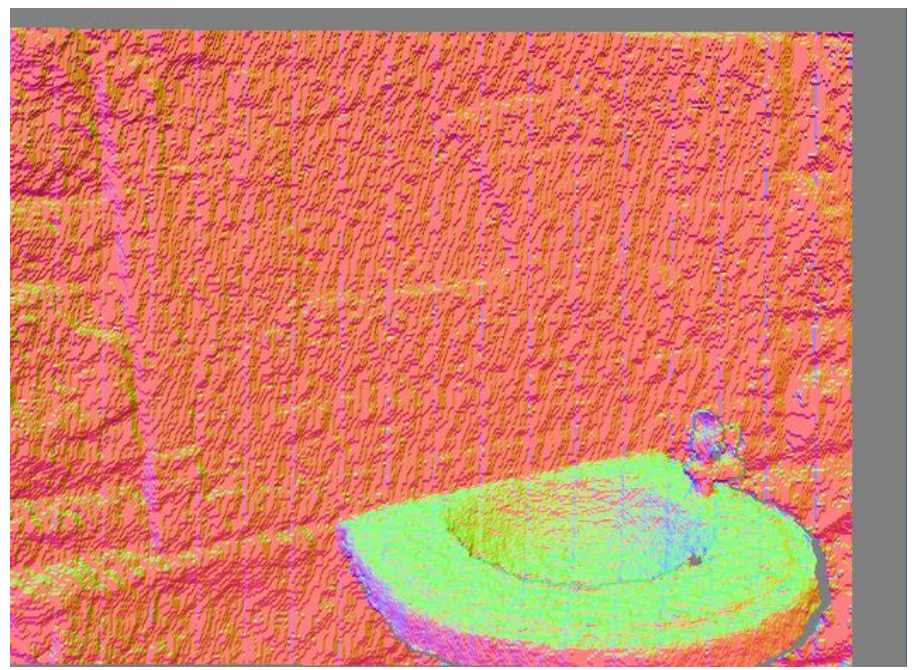


"Real-time 3D Reconstruction at Scale using Voxel Hashing", Nießner et al., ToG 2013.

State-of-the-art

Voxel Hashing

- **Baseline** RGB-D based **3D reconstruction framework**
  - initial camera poses
  - sparse SDF reconstruction
- Challenges:
  - (Slightly) **inaccurate and over-smoothed geometry**
  - **Bad colors**
  - **Inaccurate camera pose** estimation
  - **Input data** quality (e.g. motion blur, sensor noise)

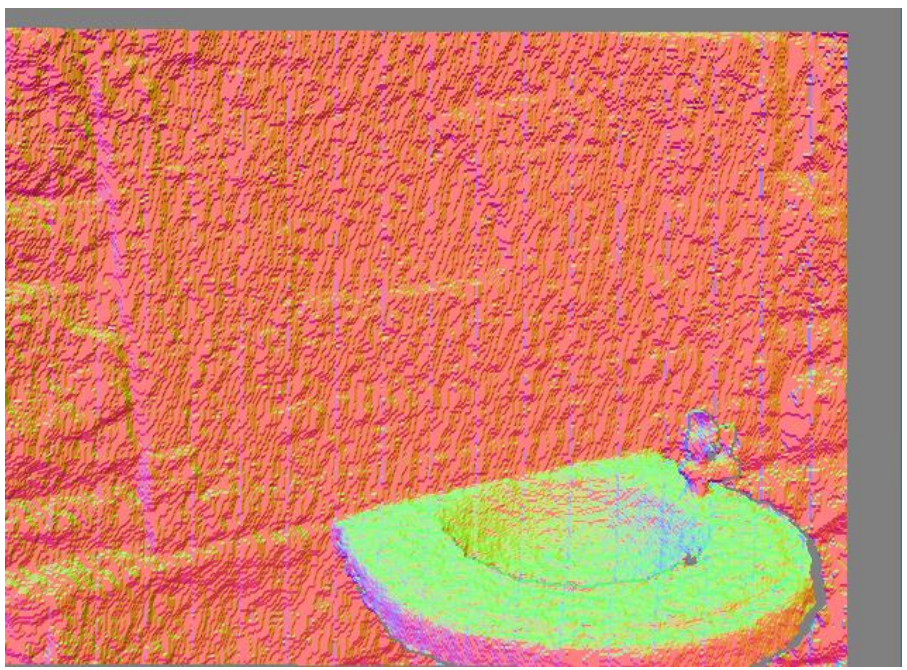


"Real-time 3D Reconstruction at Scale using Voxel Hashing", Nießner et al., ToG 2013.

State-of-the-art

Voxel Hashing

- **Baseline** RGB-D based **3D reconstruction framework**
  - initial camera poses
  - sparse SDF reconstruction
- Challenges:
  - (Slightly) **inaccurate and over-smoothed geometry**
  - **Bad colors**
  - **Inaccurate camera pose** estimation
  - **Input data** quality (e.g. motion blur, sensor noise)
- Goal: **High-Quality Reconstruction of Geometry and Color**



"Real-time 3D Reconstruction at Scale using Voxel Hashing", Nießner et al., ToG 2013.

State-of-the-art

State-of-the-art

High-Quality Colors [Zhou2014]

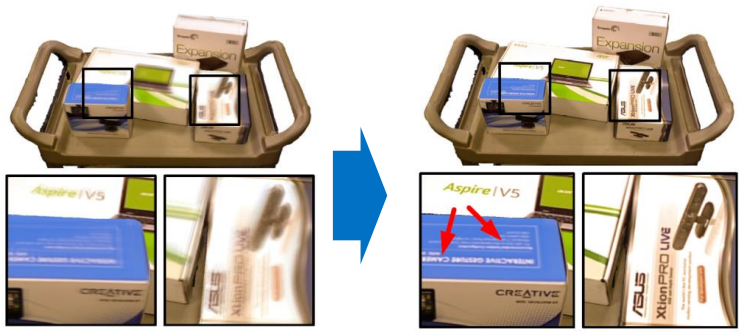
Optimize camera poses and image deformations to optimally fit initial (maybe wrong) reconstruction

But: HQ images required, no geometry refinement involved

"Color Map Optimization for 3D Reconstruction with Consumer Depth Cameras", Zhou and Koltun, ToG 2014

State-of-the-art

High-Quality Colors [Zhou2014]

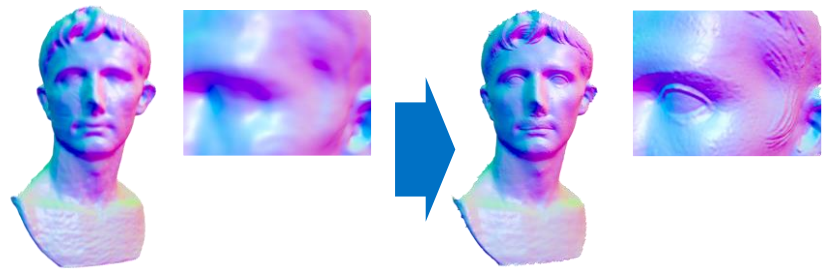


Optimize camera poses and image deformations to optimally fit initial (maybe wrong) reconstruction

But: HQ images required, no geometry refinement involved

"Color Map Optimization for 3D Reconstruction with Consumer Depth Cameras", Zhou and Koltun, ToG 2014

High-Quality Geometry [Zollhoefer2015]



Adjust camera poses in advance (bundle adjustment) to improve color

Use shading cues (RGB) to refine geometry (shading based refinement of surface & albedo)

But: RGB is fixed (no color refinement based on refined geometry)

"Shading-based Refinement on Volumetric Signed Distance Functions", Zollhoefer et al., ToG 2015

State-of-the-art

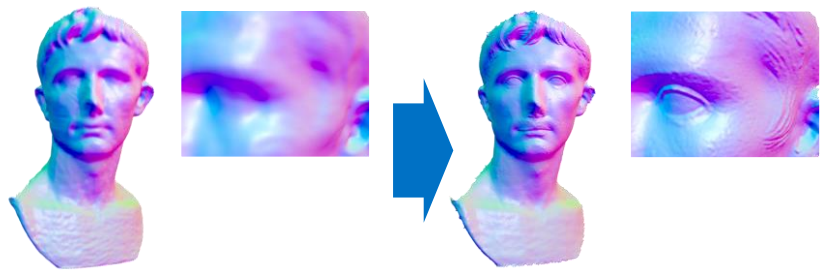
High-Quality Colors [Zhou2014]

Optimize camera poses and image deformations to optimally fit initial (maybe wrong) reconstruction

But: HQ images required, no geometry refinement involved

"Color Map Optimization for 3D Reconstruction with Consumer Depth Cameras", Zhou and Koltun, ToG 2014

High-Quality Geometry [Zollhoefer2015]



Adjust camera poses in advance (bundle adjustment) to improve color

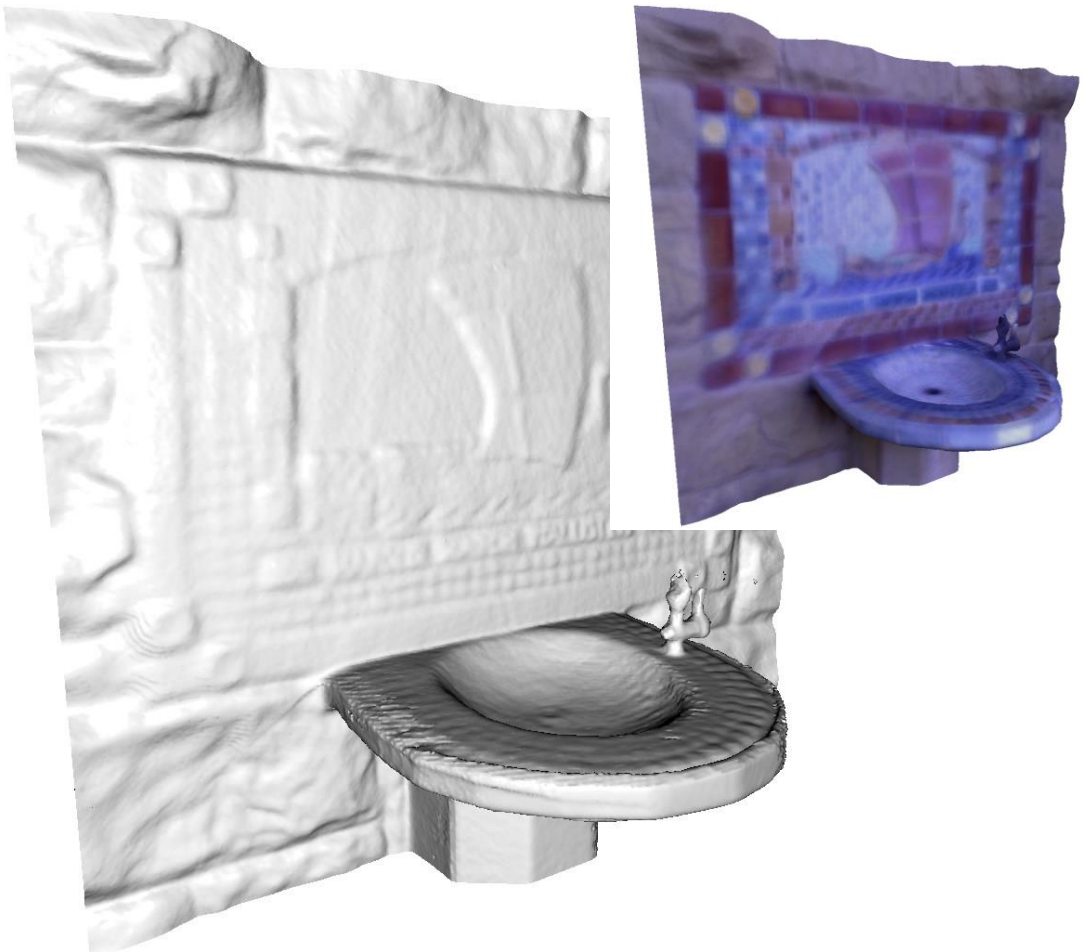
Use shading cues (RGB) to refine geometry (shading based refinement of surface & albedo)

But: RGB is fixed (no color refinement based on refined geometry)

"Shading-based Refinement on Volumetric Signed Distance Functions", Zollhoefer et al., ToG 2015

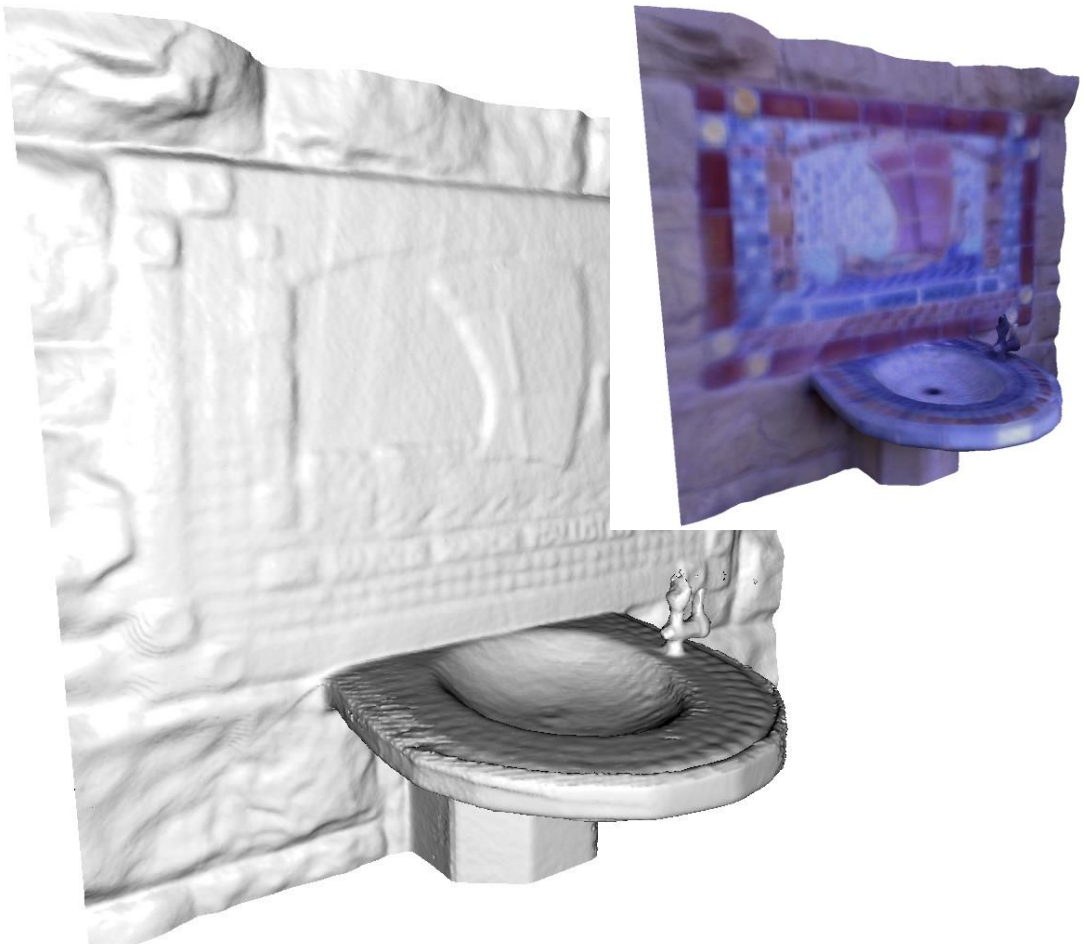
Idea: **jointly optimize for geometry, albedo and image formation model** to simultaneously obtain high-quality geometry and appearance!

Our Method



Our Method

- Temporal view **sampling & filtering** techniques (input frames)



Our Method

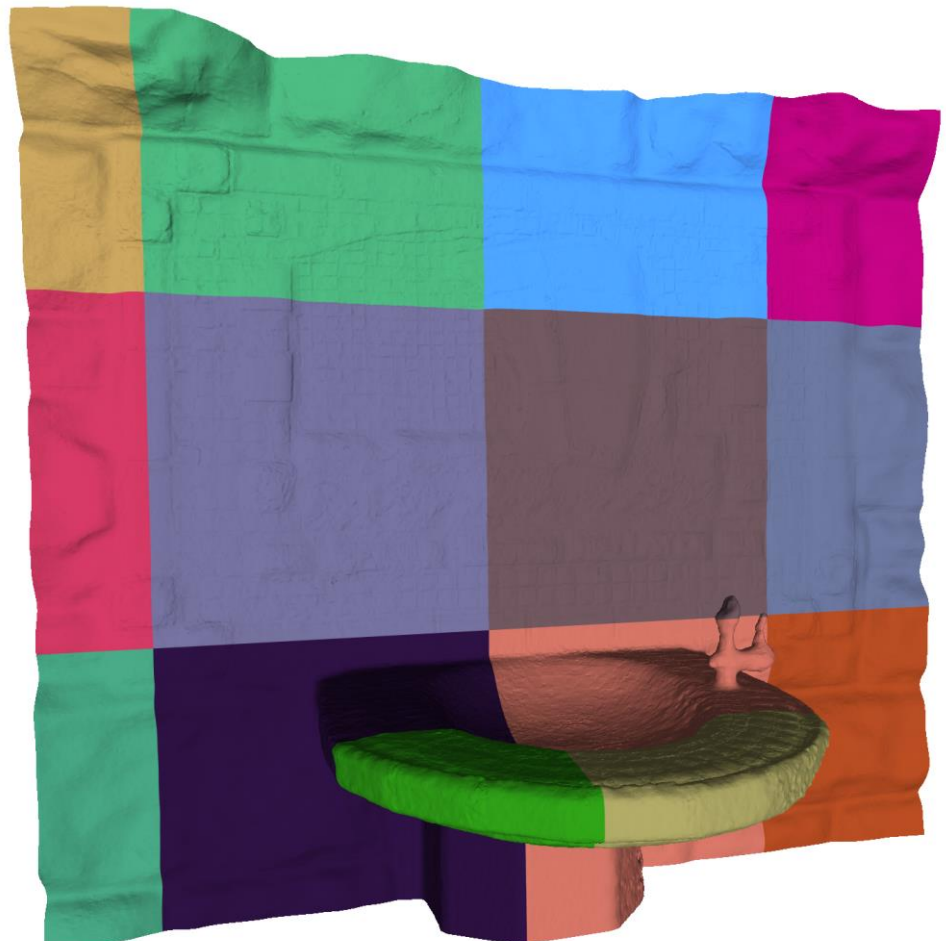
- Temporal view **sampling & filtering** techniques (input frames)
- **Joint optimization** of
  - **surface & albedo** (Signed Distance Field)
  - **image formation model**

Our Method

- Temporal view **sampling & filtering** techniques (input frames)
- **Joint optimization** of
  - **surface & albedo** (Signed Distance Field)
  - **image formation model**

Our Method

- Temporal view **sampling & filtering** techniques (input frames)
- **Joint optimization** of
  - **surface & albedo** (Signed Distance Field)
  - **image formation model**
- Lighting estimation using **Spatially-Varying Spherical Harmonics** (SVSH)



Our Method

- Temporal view **sampling & filtering** techniques (input frames)
- **Joint optimization** of
  - **surface & albedo** (Signed Distance Field)
  - **image formation model**
- Lighting estimation using **Spatially-Varying Spherical Harmonics** (SVSH)
- **Optimized colors** (by-product)

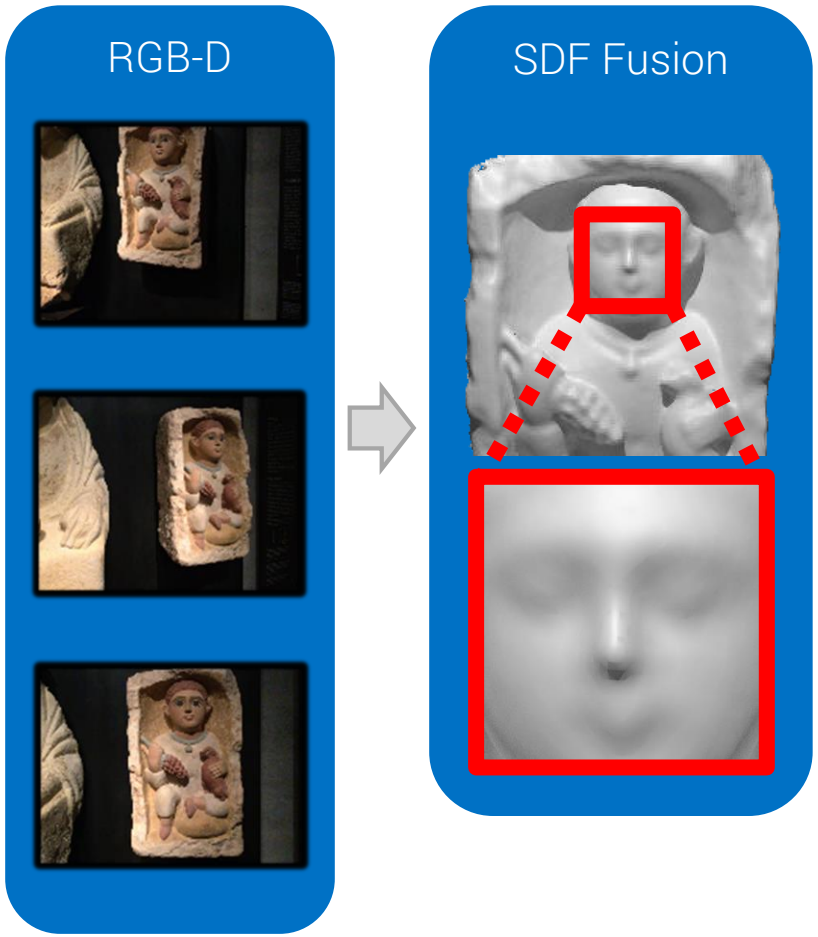
Approach

Overview

RGB-D

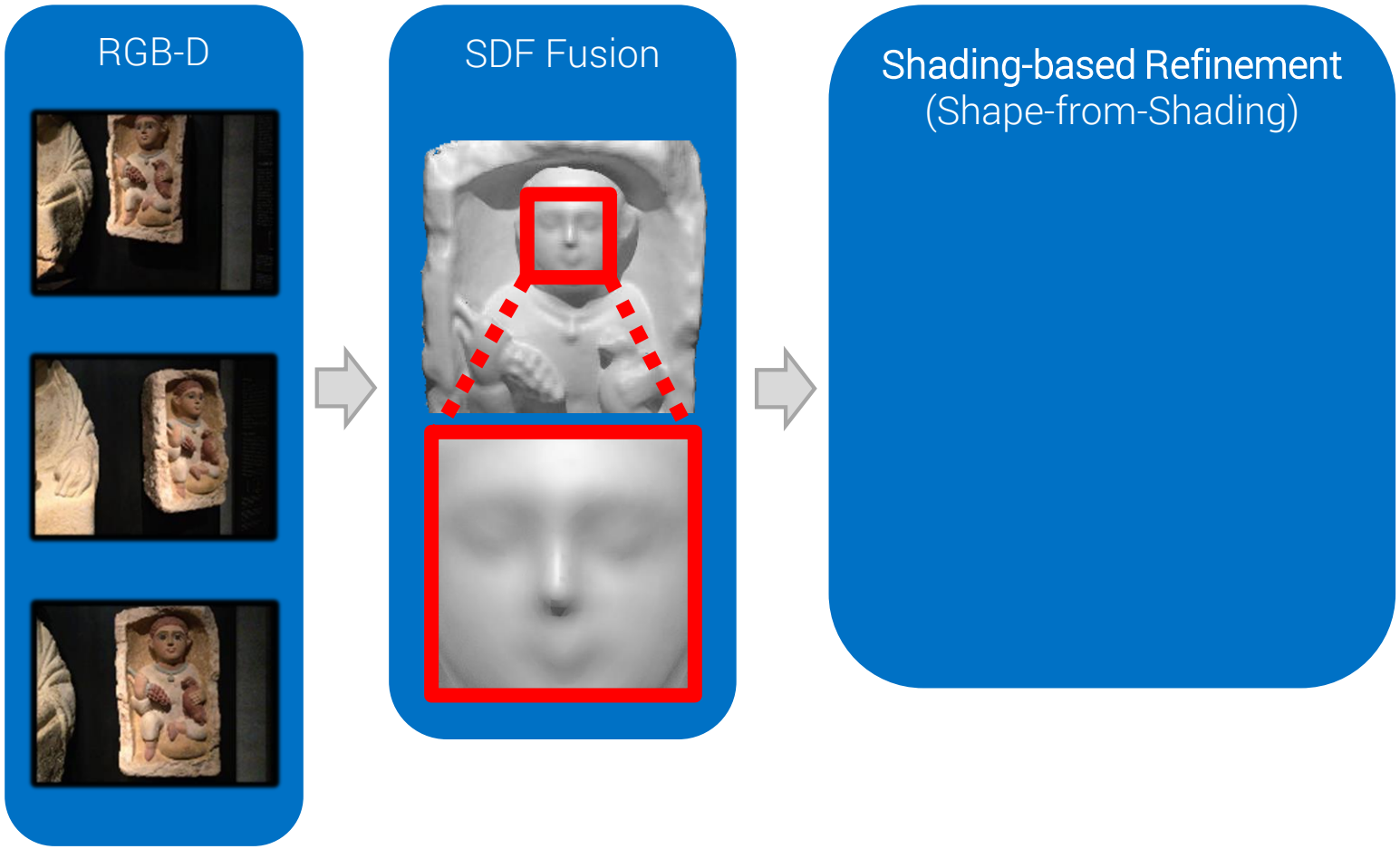
Approach

Overview



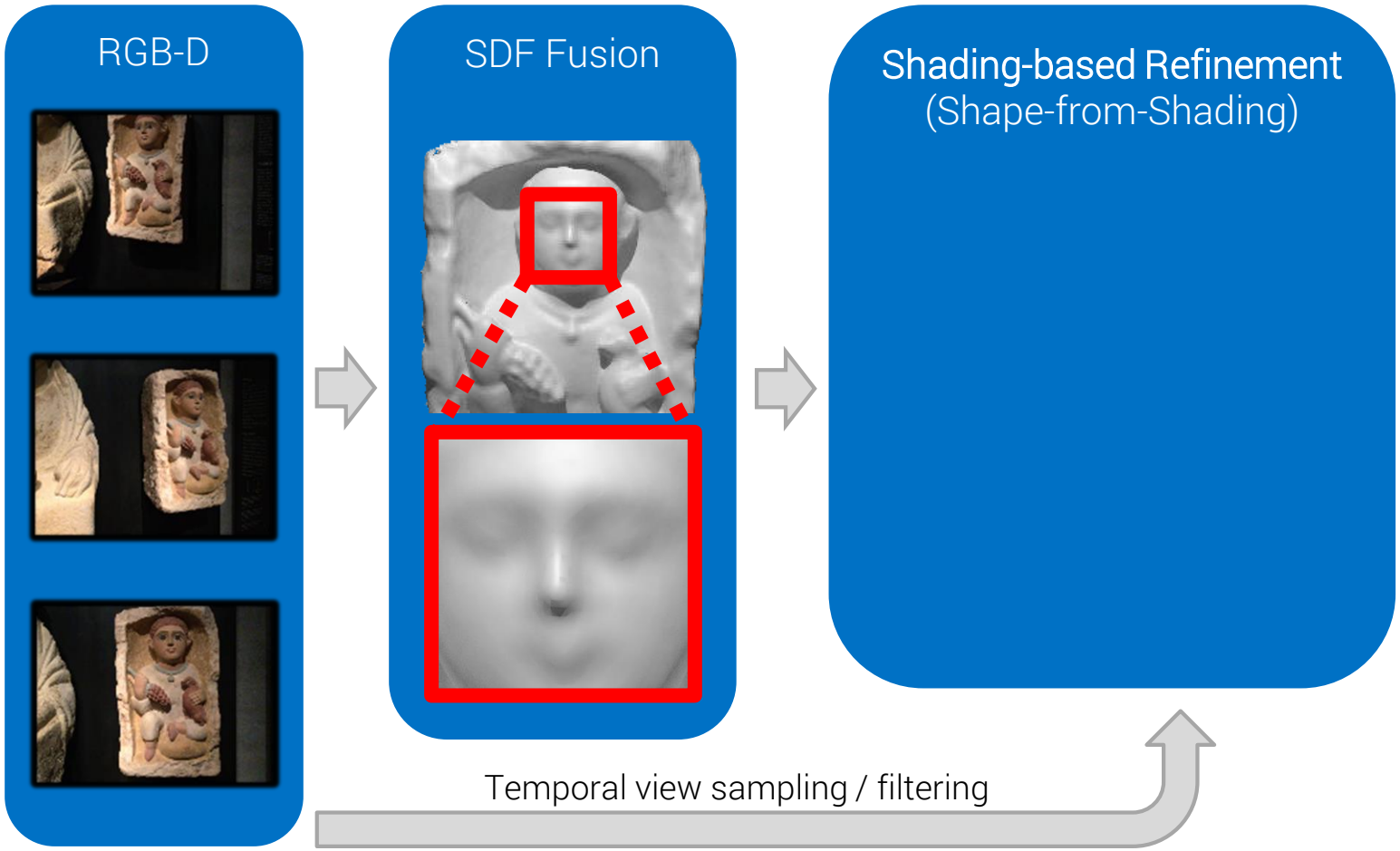
Approach

Overview



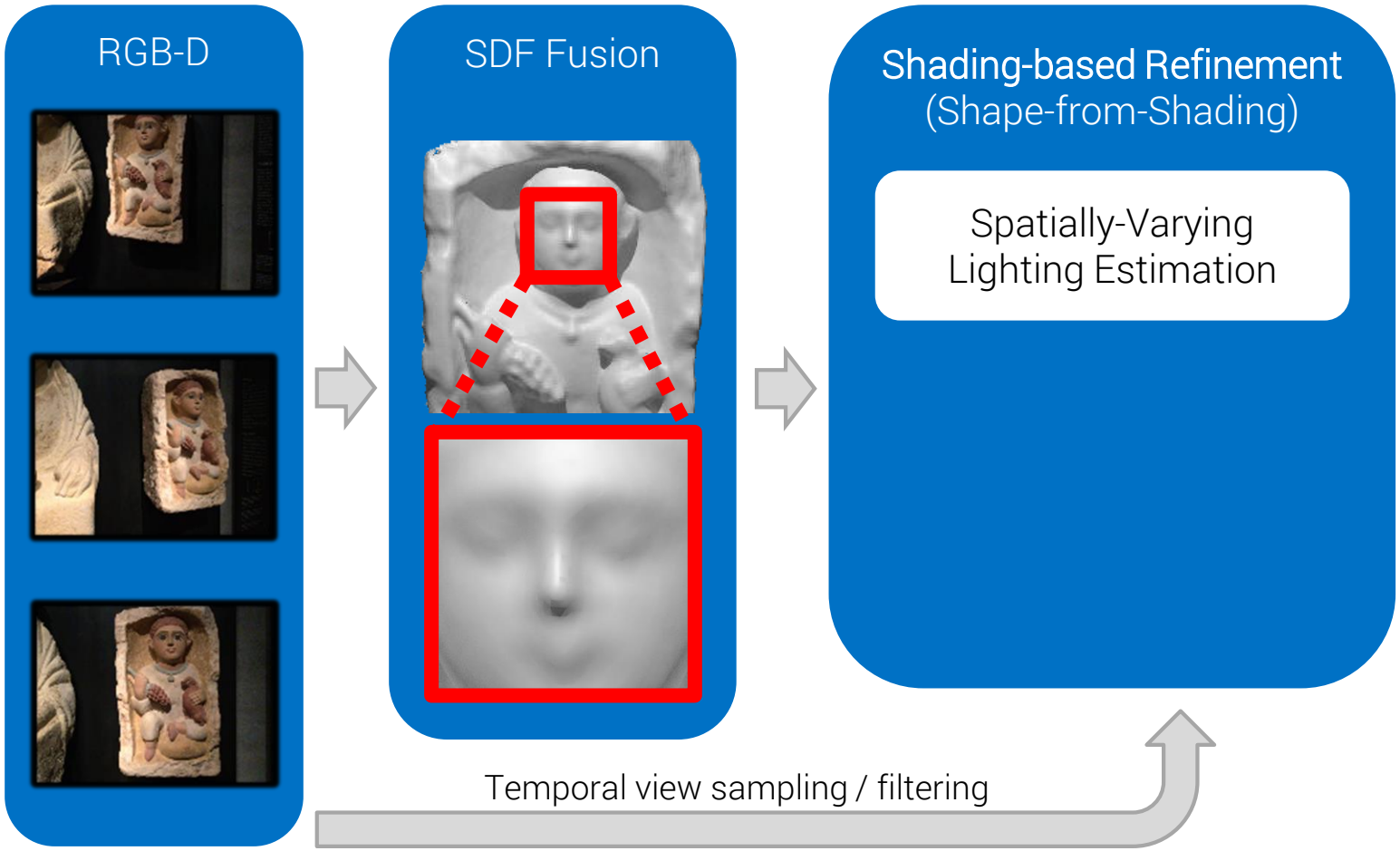
Approach

Overview



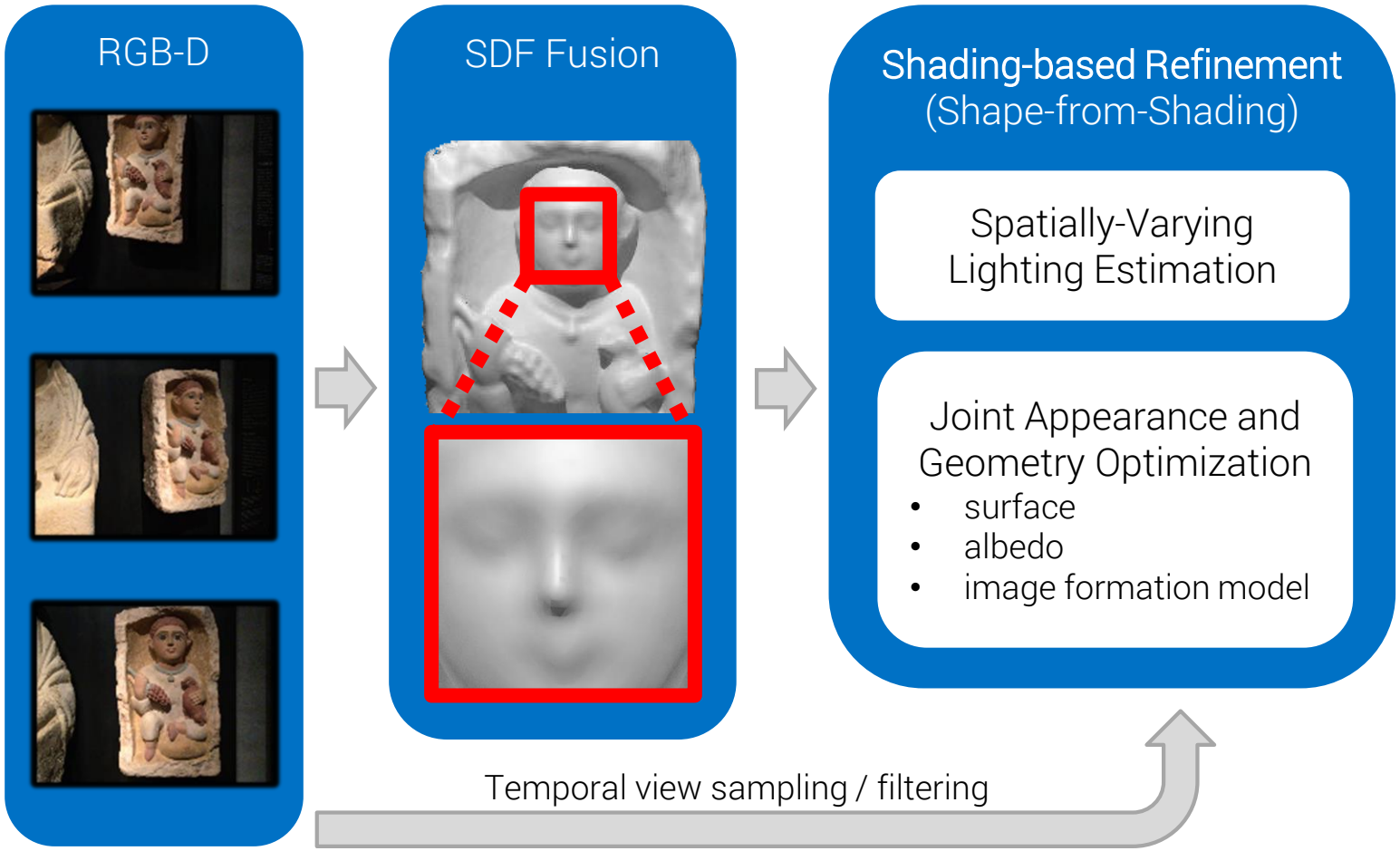
Approach

Overview



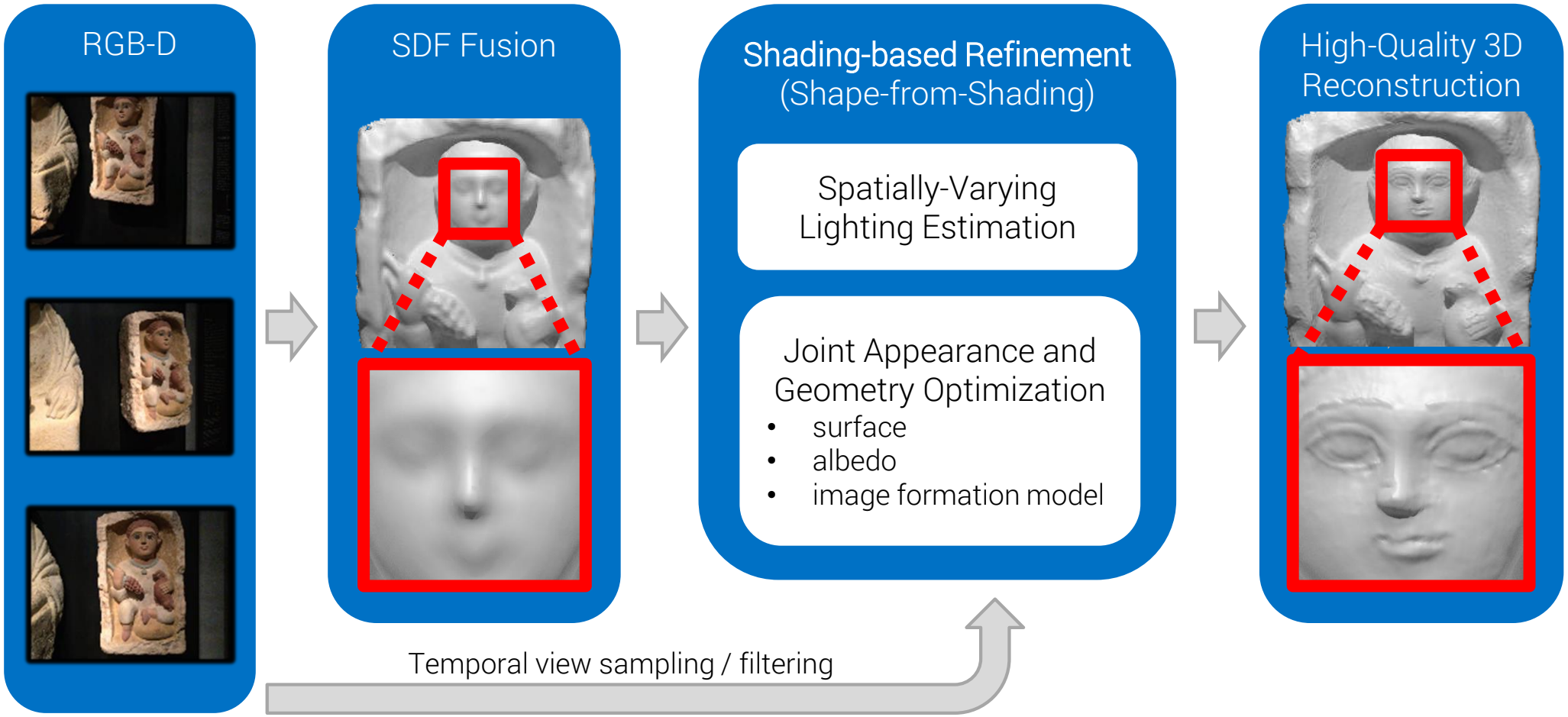
Approach

Overview



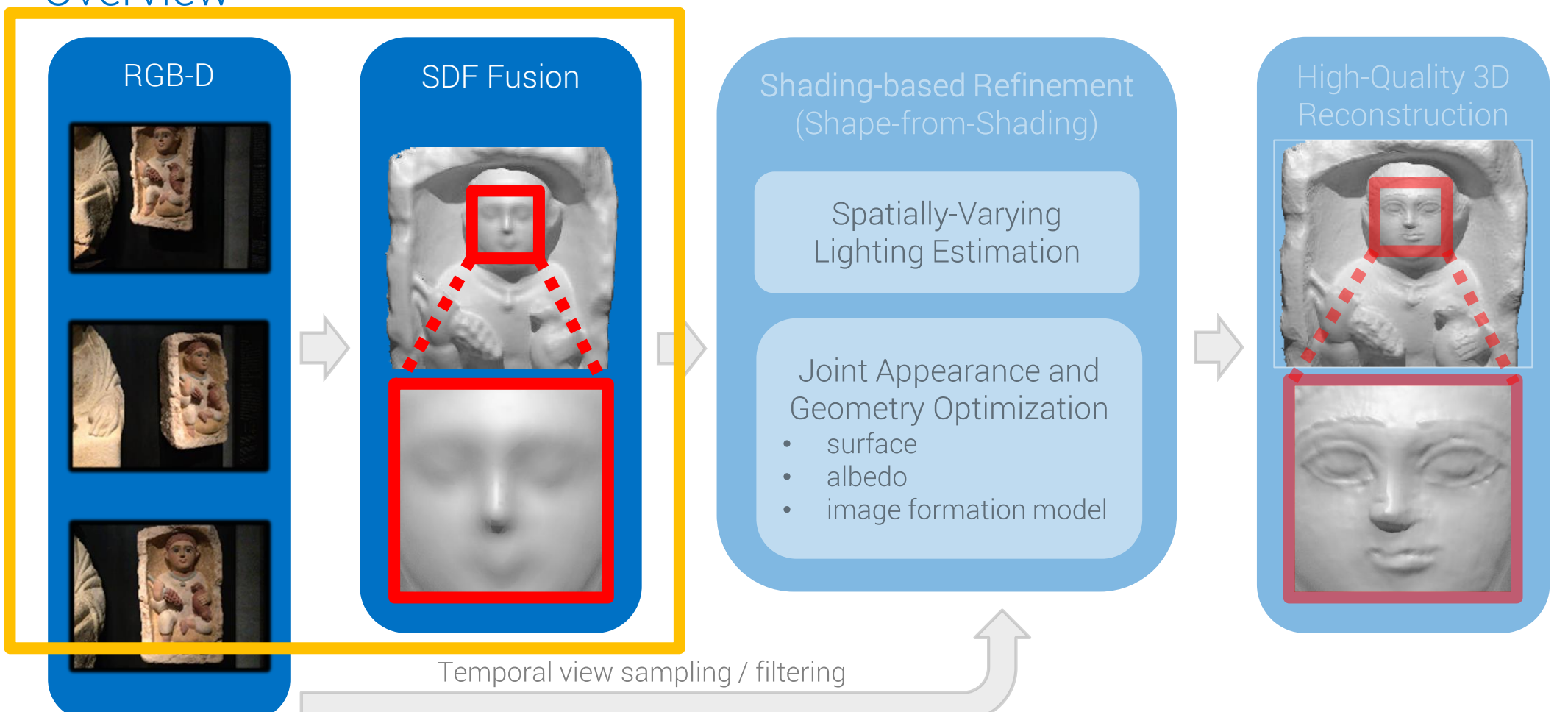
Approach

Overview



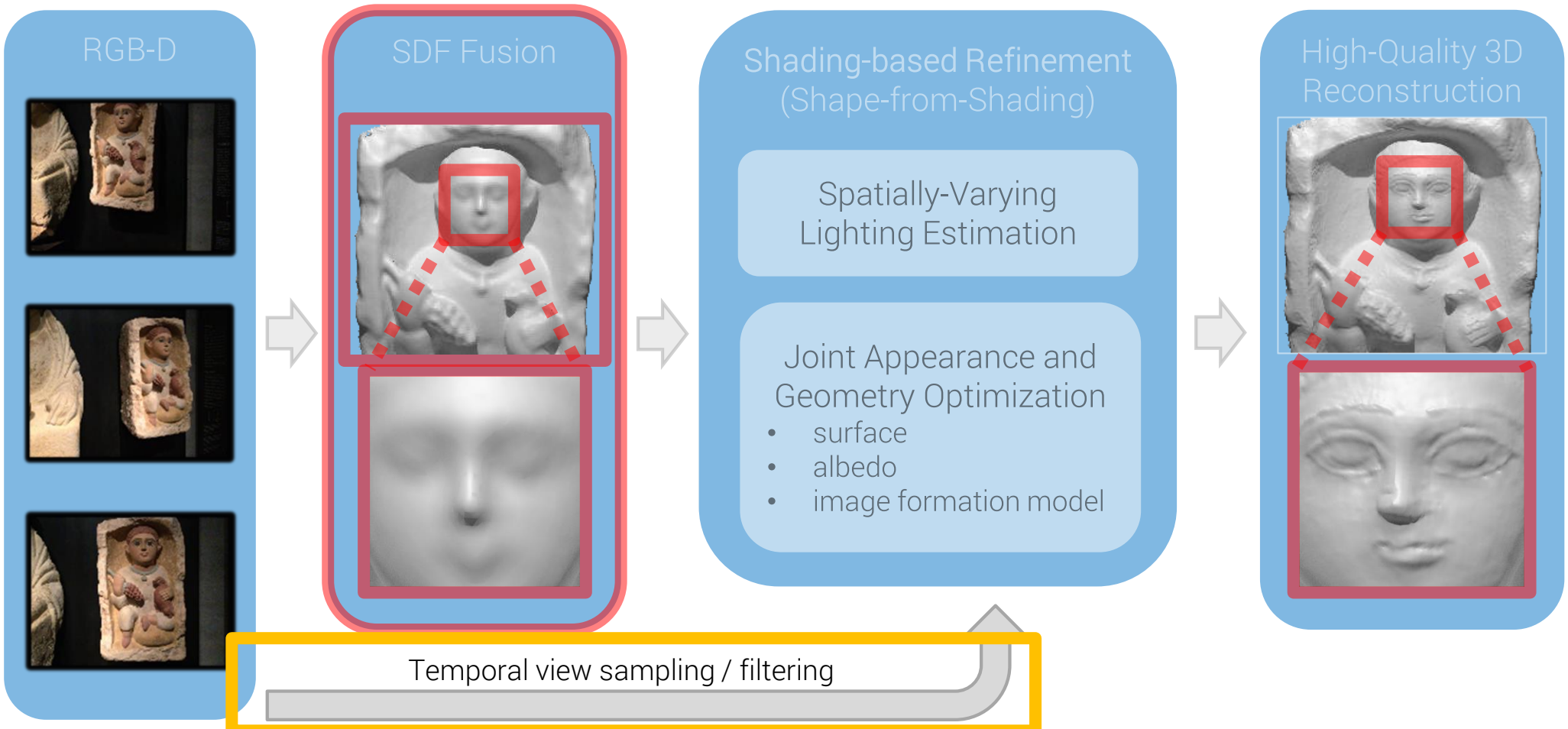
Approach

Overview



Approach

Overview



Keyframe Selection

- Compute per-frame blur score (for color image)[3]

Frame 81

Frame 84

- Select frame with best score within a fixed size window as keyframe

[3] "The blur effect: perception and estimation with a new no-reference perceptual blur metric", Crete et al., SPIE 2007.

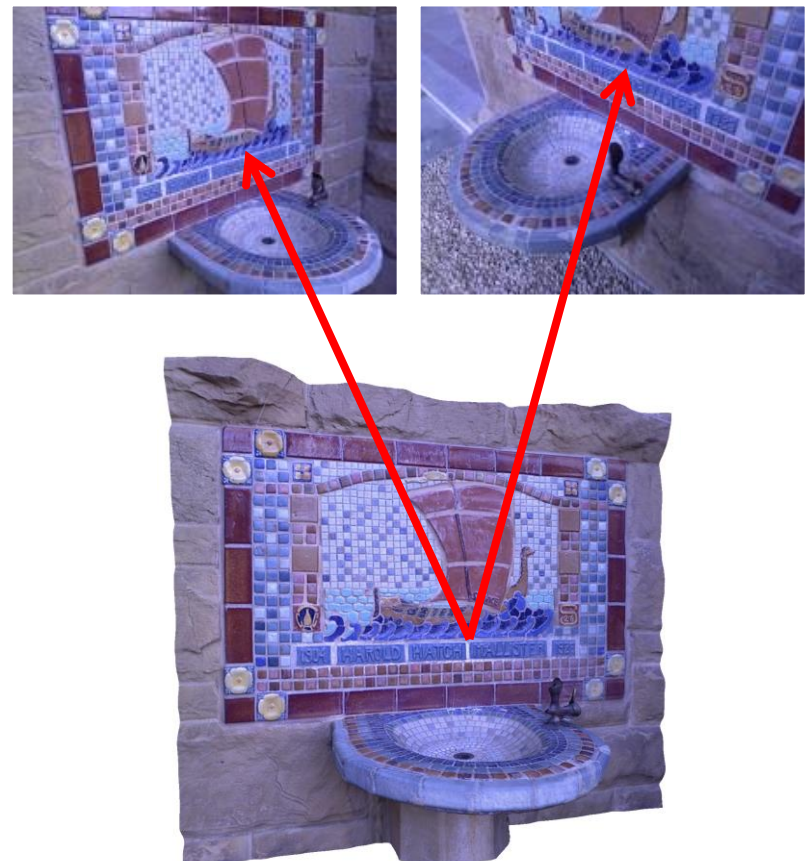
Sampling / Filtering

Sampling of voxel observations

- Sample from selected **keyframes only**
- **Collect observations** for voxel in input views:

$$c_i^v = \mathcal{C}_i(\pi(\mathcal{T}_i^{-1} \boldsymbol{v}_{\text{iso}})).$$

Input keyframes



Reconstruction

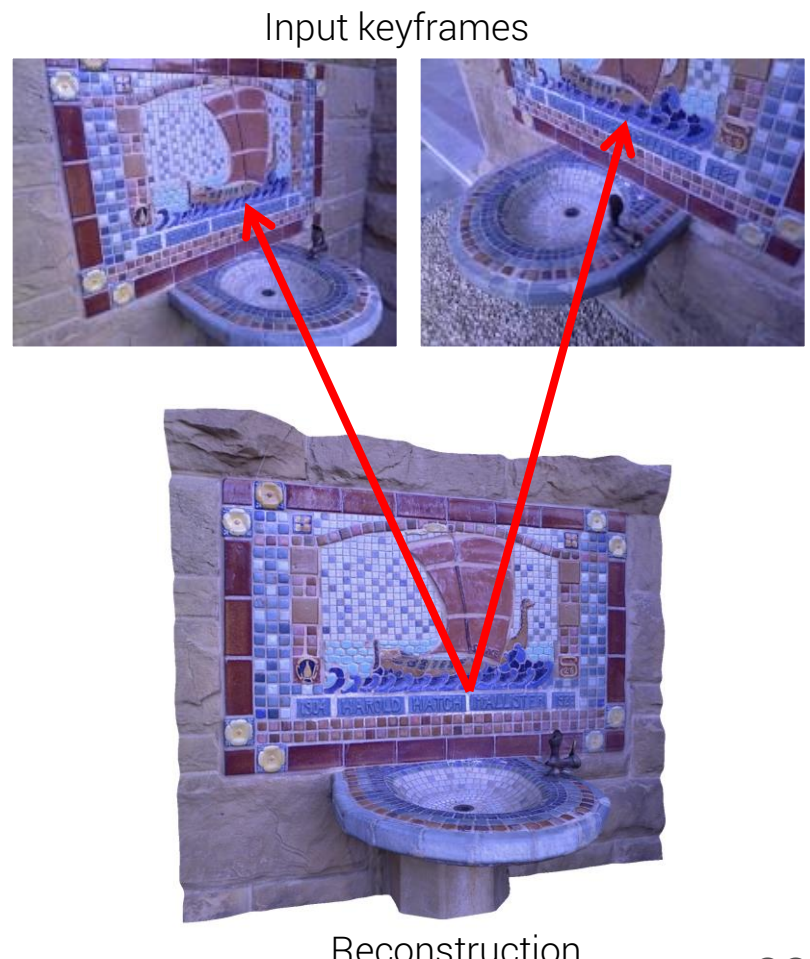
Sampling / Filtering

Sampling of voxel observations

- Sample from selected **keyframes only**
- **Collect observations** for voxel in input views:

$$c_i^v = \mathcal{C}_i(\boxed{\pi(\mathcal{T}_i^{-1} \boldsymbol{v}_{\text{iso}})}).$$

Voxel center transformed and projected into input view



Sampling / Filtering

Sampling of voxel observations

- Sample from selected **keyframes only**
- **Collect observations** for voxel in input views:

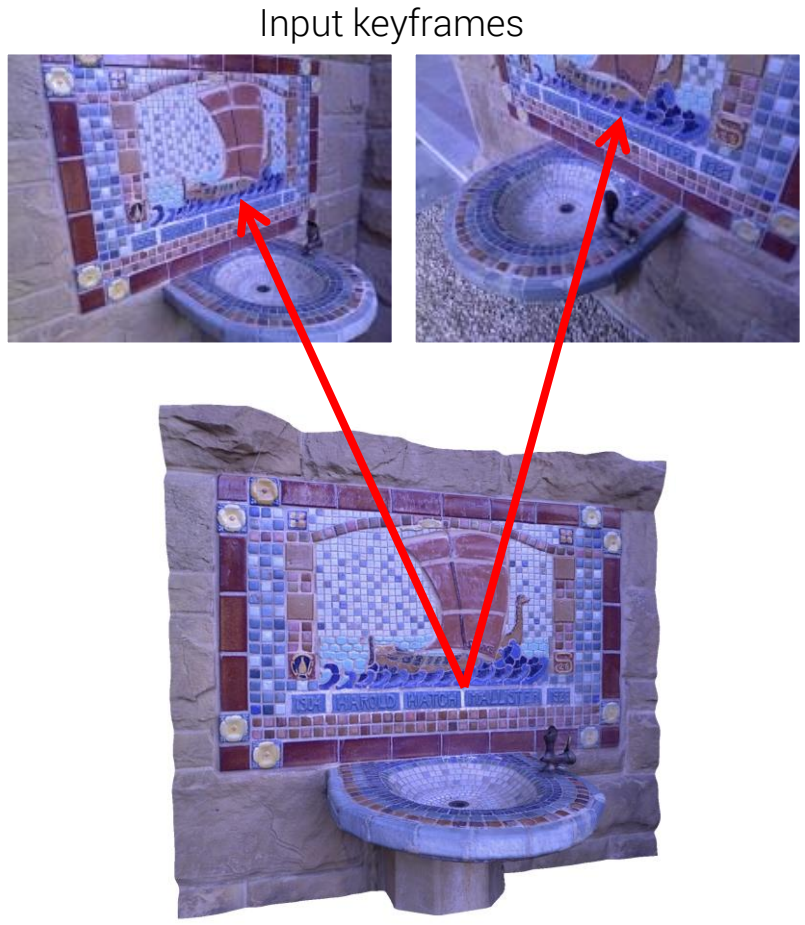
$$c_i^v = \mathcal{C}_i(\boxed{\pi(\mathcal{T}_i^{-1}\boldsymbol{v}_{\text{iso}})}).$$

Voxel center transformed and projected into input view

- Observation weights: **view-dependent** on normal and depth

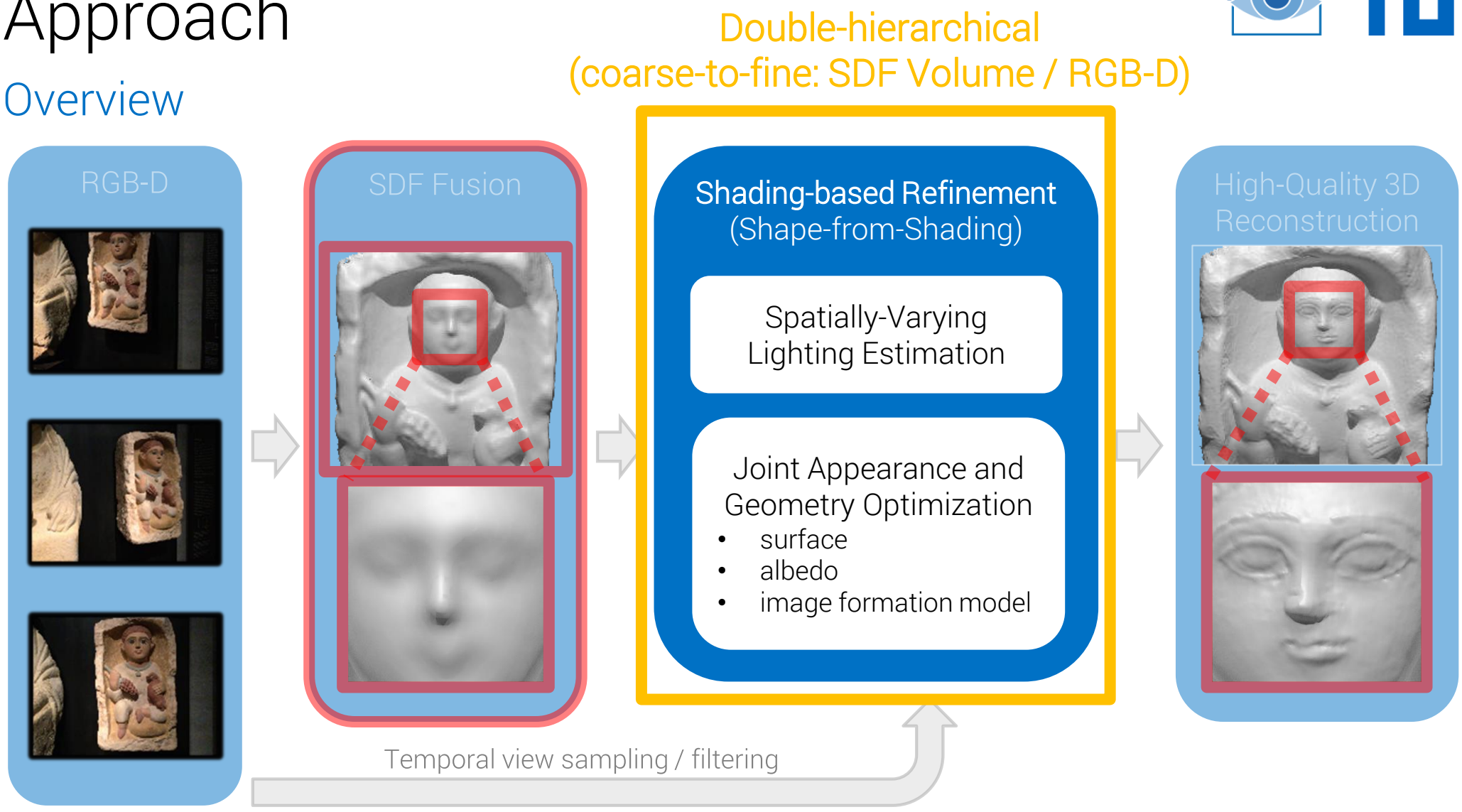
$$w_i^{\boldsymbol{v}} = \frac{\cos(\theta)}{d^2}$$

- Filter observations: keep only **best 5 observations** by weight



Approach

Overview



Shape-from-Shading

- Shading equation: $$\mathbf{B}(\boldsymbol{v}) = \mathbf{a}(\boldsymbol{v}) \sum_{m=1}^{b^2} l_m H_m(\mathbf{n}(\boldsymbol{v})),$$

Shape-from-Shading

- Shading equation:

surface normal

$$\mathbf{B}(\boldsymbol{v}) = \mathbf{a}(\boldsymbol{v}) \sum_{m=1}^{b^2} l_m H_m(\boxed{\mathbf{n}(\boldsymbol{v})}),$$

Shape-from-Shading

- Shading equation:

surface normal

$$\mathbf{B}(\boldsymbol{v}) = \mathbf{a}(\boldsymbol{v}) \sum_{m=1}^{b^2} \boxed{l_m} H_m \boxed{(\mathbf{n}(\boldsymbol{v}))},$$

lighting

Shape-from-Shading

- Shading equation:

albedo surface normal

$$\mathbf{B}(\boldsymbol{v}) = \mathbf{a}(\boldsymbol{v}) \sum_{m=1}^{b^2} l_m H_m(\mathbf{n}(\boldsymbol{v})),$$

lighting

Shape-from-Shading

- Shading equation:

Shading albedo surface normal

$$\mathbf{B}(\boldsymbol{v}) = \mathbf{a}(\boldsymbol{v}) \sum_{m=1}^{b^2} l_m H_m(\mathbf{n}(\boldsymbol{v})),$$

lighting

Shape-from-Shading

- Shading equation:

Shading albedo surface normal

$$\mathbf{B}(\boldsymbol{v}) = \mathbf{a}(\boldsymbol{v}) \sum_{m=1}^{b^2} l_m H_m(\mathbf{n}(\boldsymbol{v})),$$

lighting

- Shading-based refinement:
  - Intuition: high-frequency changes in surface geometry → shading cues in input images

Shape-from-Shading

- Shading equation:

Shading albedo surface normal

$$\mathbf{B}(\boldsymbol{v}) = \mathbf{a}(\boldsymbol{v}) \sum_{m=1}^{b^2} l_m H_m(\mathbf{n}(\boldsymbol{v})),$$

lighting

- Shading-based refinement:
  - Intuition: high-frequency changes in surface geometry → shading cues in input images
  - Estimate lighting given surface and albedo (intrinsic material properties)

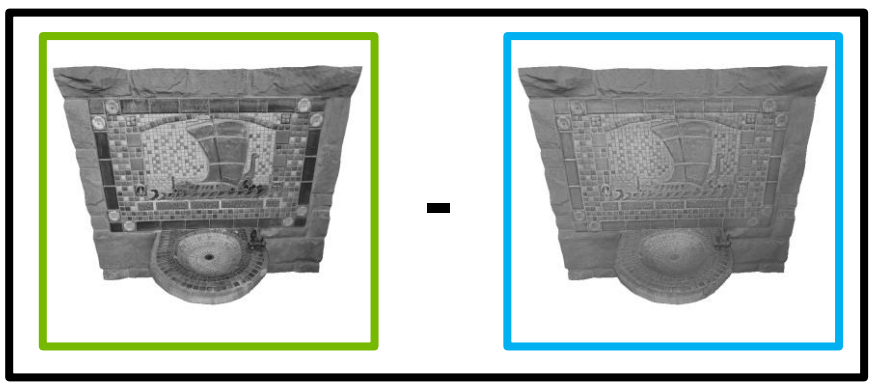
Shape-from-Shading

- Shading equation:

Shading albedo surface normal

$$\mathbf{B}(\boldsymbol{v}) = \mathbf{a}(\boldsymbol{v}) \sum_{m=1}^{b^2} l_m H_m(\mathbf{n}(\boldsymbol{v})),$$

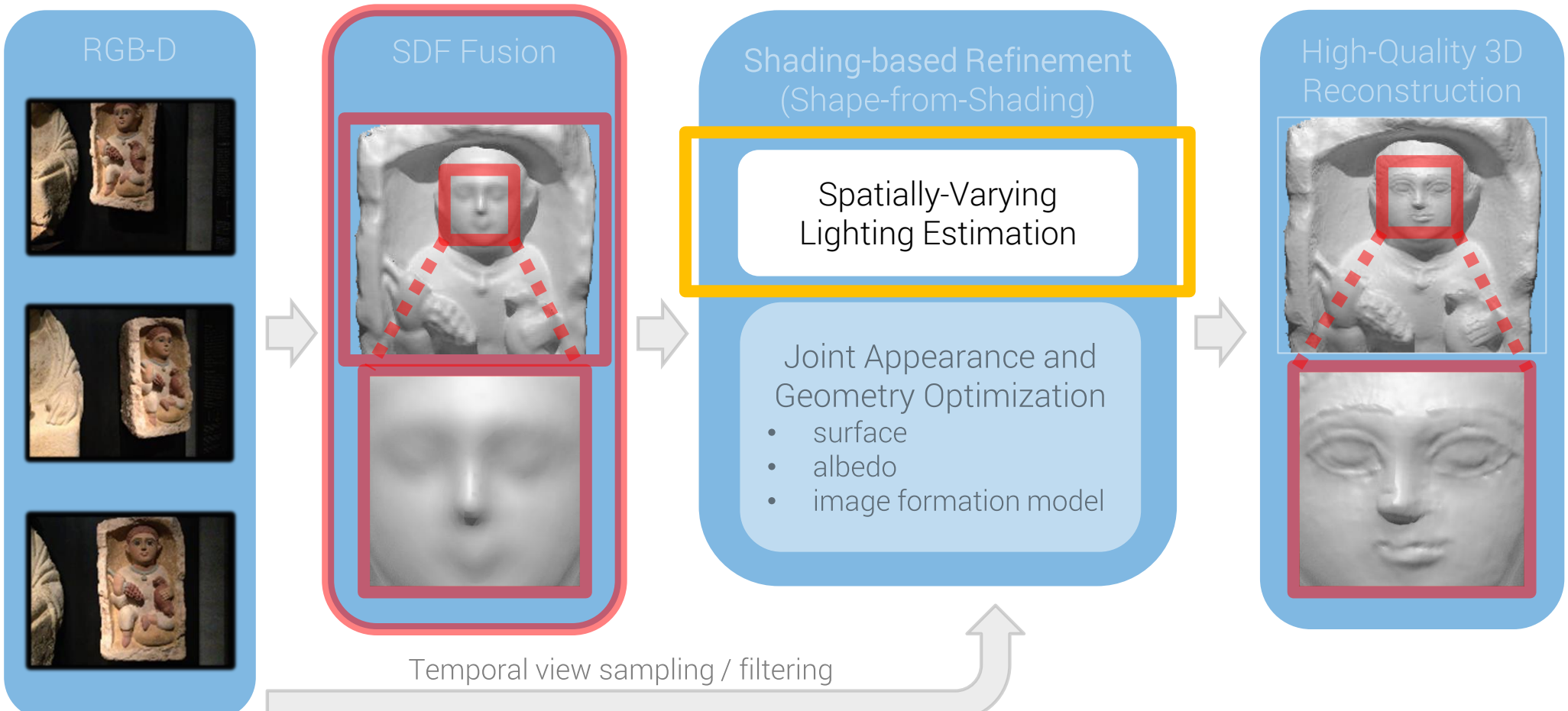
lighting



- Shading-based refinement:
  - Intuition: high-frequency changes in surface geometry → shading cues in input images
  - Estimate lighting given surface and albedo (intrinsic material properties)
  - Estimate surface and albedo given the lighting: minimize difference between estimated shading and input luminance

Approach

Overview



Lighting Estimation

Spherical Harmonics (SH)

- **Encode incident lighting** for a given surface point
- **Smooth** for Lambertian surfaces

Lighting Estimation

Spherical Harmonics (SH)

- **Encode incident lighting** for a given surface point
- **Smooth** for Lambertian surfaces
- SH Basis functions H_m parametrized by **unit normal** n

$$\mathbf{B}(\boldsymbol{v}) = \mathbf{a}(\boldsymbol{v}) \sum_{m=1}^{b^2} l_m H_m(\mathbf{n}(\boldsymbol{v}))$$

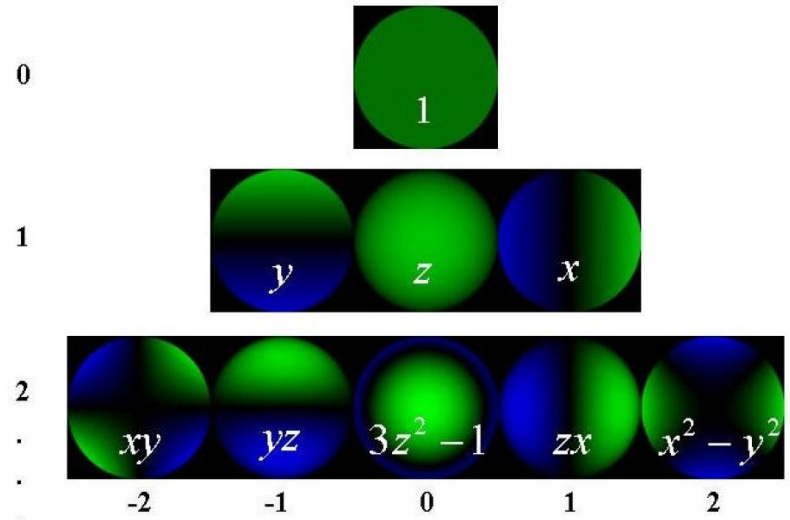
Lighting Estimation

Spherical Harmonics (SH)

- **Encode incident lighting** for a given surface point
- **Smooth** for Lambertian surfaces
- SH Basis functions H_m parametrized by **unit normal** n

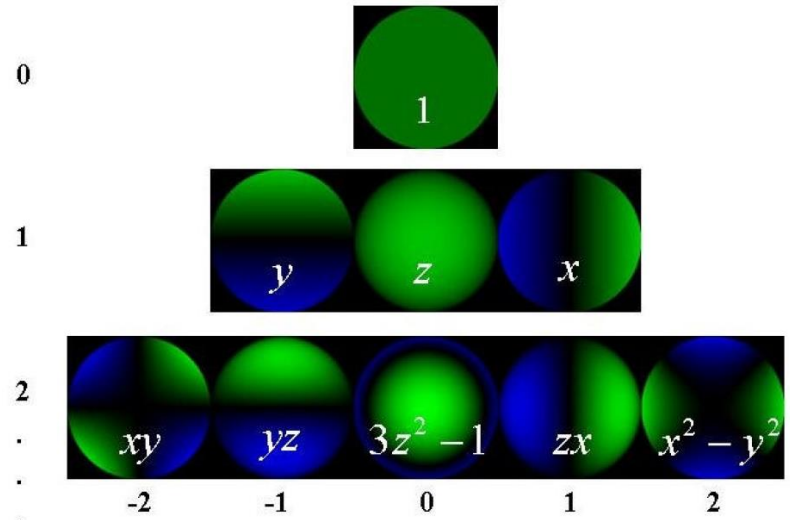
$$\mathbf{B}(\boldsymbol{v}) = \mathbf{a}(\boldsymbol{v}) \sum_{m=1}^{b^2} l_m H_m(\mathbf{n}(\boldsymbol{v}))$$

- **Good approx.** using **only 9 SH basis functions** (2nd order)



Lighting Estimation

Spherical Harmonics (SH)



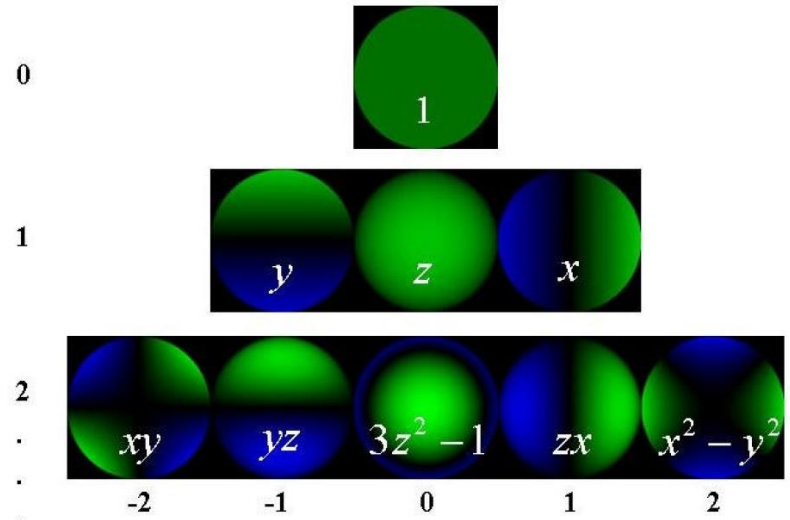
- **Encode incident lighting** for a given surface point
- **Smooth** for Lambertian surfaces
- SH Basis functions H_m parametrized by **unit normal** n

$$\mathbf{B}(\boldsymbol{v}) = \mathbf{a}(\boldsymbol{v}) \sum_{m=1}^{b^2} l_m H_m(\mathbf{n}(\boldsymbol{v}))$$

- **Good approx.** using **only 9 SH basis functions** (2nd order)
- Estimate SH coefficients: $E_{\text{light}}(\mathbf{l}) = \sum_{\mathbf{v} \in \mathbf{D}_0} (B(\mathbf{v}) - \mathbf{I}(\mathbf{v}))^2$

Lighting Estimation

Spherical Harmonics (SH)



- **Encode incident lighting** for a given surface point
- **Smooth** for Lambertian surfaces
- SH Basis functions H_m parametrized by **unit normal** n

$$\mathbf{B}(\boldsymbol{v}) = \mathbf{a}(\boldsymbol{v}) \sum_{m=1}^{b^2} l_m H_m(\mathbf{n}(\boldsymbol{v}))$$

- **Good approx.** using **only 9 SH basis functions** (2nd order)
- Estimate SH coefficients: $E_{\text{light}}(\mathbf{l}) = \sum_{\mathbf{v} \in \mathbf{D}_0} (B(\mathbf{v}) - \mathbf{I}(\mathbf{v}))^2$

- **Shortcoming**: purely directional → cannot represent scene lighting for all surface points simultaneously

Spatially-Varying Lighting

Subvolume Partitioning

Spatially-Varying Lighting

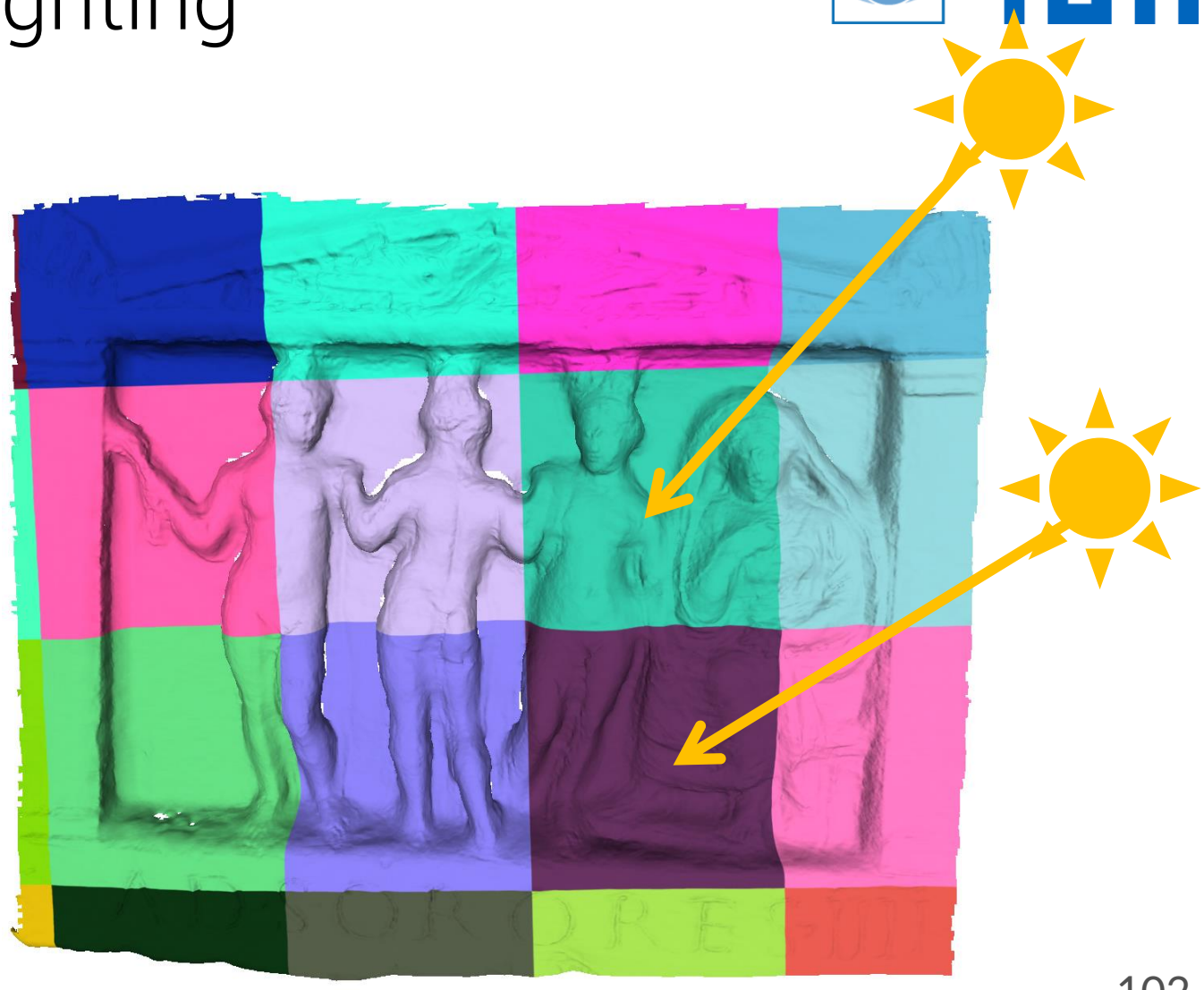
Subvolume Partitioning

- **Partition SDF** volume into subvolumes

Spatially-Varying Lighting

Subvolume Partitioning

- **Partition SDF** volume into subvolumes
- Estimate **independent SH coefficients** for each **subvolume**



Spatially-Varying Lighting

Subvolume Partitioning

- **Partition SDF** volume into subvolumes
- Estimate **independent SH coefficients** for each **subvolume**
- Obtain **per-voxel SH coefficients** through tri-linear interpolation

Spatially-Varying Lighting

Joint Optimization

Spatially-Varying Lighting

Joint Optimization

- Estimate SVSH coefficients for all subvolumes jointly:

$$E_{\text{lighting}}(\boldsymbol{l}_1, \ldots, \boldsymbol{l}_K) = E_{\text{appearance}} + \lambda_{\text{diffuse}} E_{\text{diffuse}}.$$

Spatially-Varying Lighting

Joint Optimization

- Estimate SVSH coefficients for all subvolumes jointly:

$$E_{\text{lighting}}(\boldsymbol{l}_1, \ldots, \boldsymbol{l}_K) = \boxed{E_{\text{appearance}}} + \lambda_{\text{diffuse}} E_{\text{diffuse}}.$$

Data term:

$$E_{\text{appearance}} = \sum_{\boldsymbol{v} \in \mathbf{D}_0} (\mathbf{B}(\boldsymbol{v}) - \mathbf{I}(\boldsymbol{v}))^2.$$

Similarity between estimated shading and input luminance

Spatially-Varying Lighting

Joint Optimization

- Estimate SVSH coefficients for all subvolumes jointly:

$$E_{\text{lighting}}(\boldsymbol{l}_1, \ldots, \boldsymbol{l}_K) = \boxed{E_{\text{appearance}}} + \lambda_{\text{diffuse}} \boxed{E_{\text{diffuse}}}.$$

Data term:

$$E_{\text{appearance}} = \sum_{\boldsymbol{v} \in \mathbf{D}_0} (\mathbf{B}(\boldsymbol{v}) - \mathbf{I}(\boldsymbol{v}))^2.$$

Similarity between estimated shading and input luminance

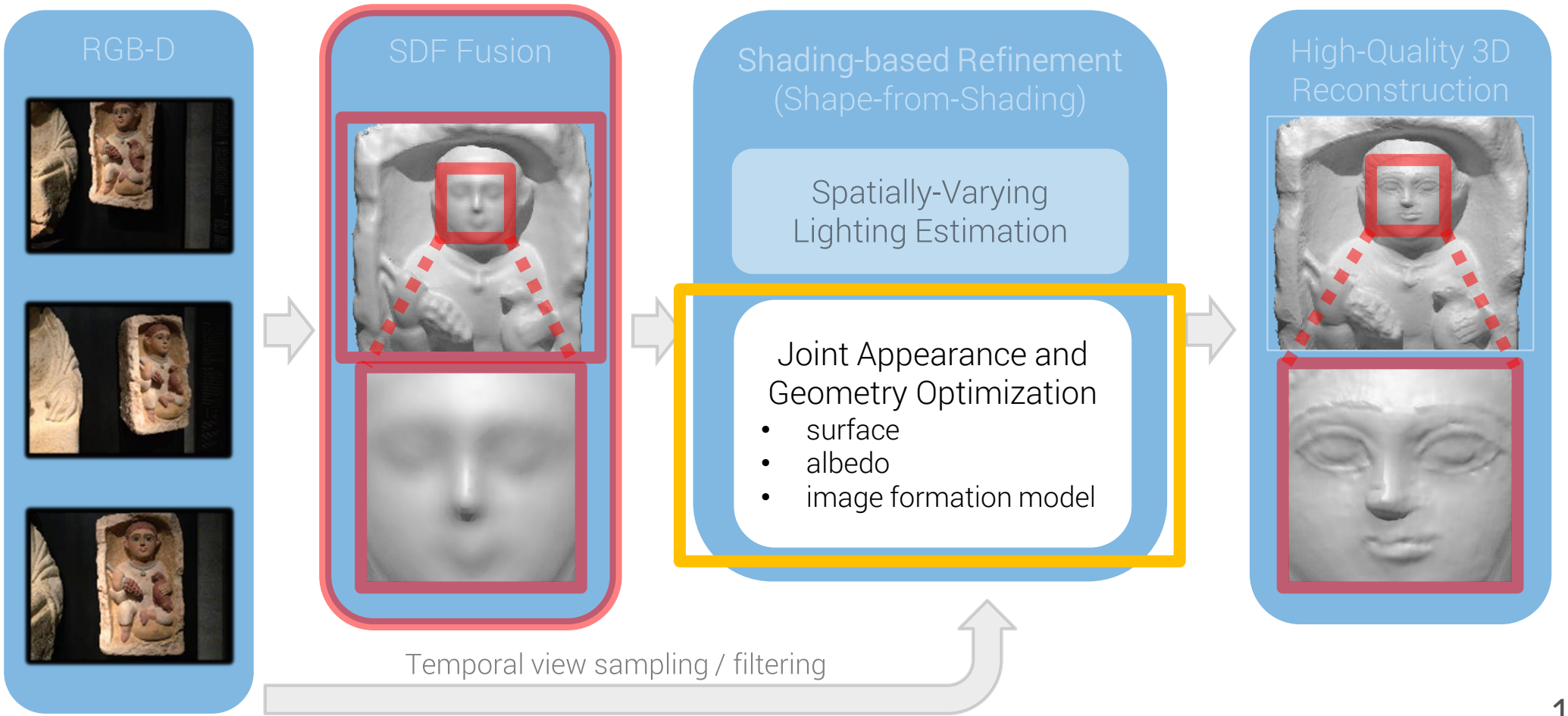
Laplacian regularizer:

$$E_{\text{diffuse}} = \sum_{s \in \mathcal{S}} \sum_{r \in \mathcal{N}_s} (\boldsymbol{l}_s - \boldsymbol{l}_r)^2.$$

Smooth illumination changes

Approach

Overview



Joint Optimization

Shading-based SDF optimization

- **Joint optimization** of geometry, albedo and image formation model (camera poses and camera intrinsics):

$$E_{\text{scene}}(\mathcal{X}) = \sum_{\boldsymbol{v} \in \tilde{\mathbf{D}}_0} \lambda_g E_g + \lambda_v E_v + \lambda_s E_s + \lambda_a E_a$$

with $\mathcal{X} = (\mathcal{T}, \tilde{\mathbf{D}}, \mathbf{a}, f_x, f_y, c_x, c_y, \kappa_1, \kappa_2, \rho_1)$

Joint Optimization

Shading-based SDF optimization

- **Joint optimization** of geometry, albedo and image formation model (camera poses and camera intrinsics):

$$E_{\text{scene}}(\mathcal{X}) = \sum_{\boldsymbol{v} \in \tilde{\mathbf{D}}_0} \lambda_g \boxed{E_g} + \lambda_v E_v + \lambda_s E_s + \lambda_a E_a$$

with $\mathcal{X} = (\mathcal{T}, \tilde{\mathbf{D}}, \mathbf{a}, f_x, f_y, c_x, c_y, \kappa_1, \kappa_2, \rho_1)$

Gradient-based shading constraint (data term)

Joint Optimization

Shading-based SDF optimization

- **Joint optimization** of geometry, albedo and image formation model (camera poses and camera intrinsics):

$$E_{\text{scene}}(\mathcal{X}) = \sum_{\boldsymbol{v} \in \tilde{\mathbf{D}}_0} \lambda_g \boxed{E_g} + \lambda_v \boxed{E_v} + \lambda_s E_s + \lambda_a E_a$$

with $\mathcal{X} = (\mathcal{T}, \tilde{\mathbf{D}}, \mathbf{a}, f_x, f_y, c_x, c_y, \kappa_1, \kappa_2, \rho_1)$

Gradient-based shading constraint (data term)

Volumetric regularizer: smoothness in distance values (Laplacian)

$$E_v(\boldsymbol{v}) = (\Delta \tilde{\mathbf{D}}(\boldsymbol{v}))^2$$

Joint Optimization

Shading-based SDF optimization

- **Joint optimization** of geometry, albedo and image formation model (camera poses and camera intrinsics):

$$E_{\text{scene}}(\mathcal{X}) = \sum_{\boldsymbol{v} \in \tilde{\mathbf{D}}_0} \lambda_g \boxed{E_g} + \lambda_v \boxed{E_v} + \lambda_s \boxed{E_s} + \lambda_a E_a$$

with $\mathcal{X} = (\mathcal{T}, \tilde{\mathbf{D}}, \mathbf{a}, f_x, f_y, c_x, c_y, \kappa_1, \kappa_2, \rho_1)$

Gradient-based shading constraint (data term)

Volumetric regularizer: smoothness in distance values (Laplacian)

Surface Stabilization constraint: stay close to initial distance values

$$E_s(\boldsymbol{v}) = (\tilde{\mathbf{D}}(\boldsymbol{v}) - \mathbf{D}(\boldsymbol{v}))^2$$

Joint Optimization

Shading-based SDF optimization

- **Joint optimization** of geometry, albedo and image formation model (camera poses and camera intrinsics):

$$E_{\text{scene}}(\mathcal{X}) = \sum_{\boldsymbol{v} \in \tilde{\mathbf{D}}_0} \lambda_g \boxed{E_g} + \lambda_v \boxed{E_v} + \lambda_s \boxed{E_s} + \lambda_a \boxed{E_a}$$

with $\mathcal{X} = (\mathcal{T}, \tilde{\mathbf{D}}, \mathbf{a}, f_x, f_y, c_x, c_y, \kappa_1, \kappa_2, \rho_1)$

Gradient-based shading constraint (data term)

Volumetric regularizer: smoothness in distance values (Laplacian)

Surface Stabilization constraint: stay close to initial distance values

Albedo regularizer: constrain albedo changes based on chromaticity (Laplacian)

$$E_a(\boldsymbol{v}) = \sum_{\boldsymbol{u} \in \mathcal{N}_{\boldsymbol{v}}} \phi(\mathbf{\Gamma}(\boldsymbol{v}) - \mathbf{\Gamma}(\boldsymbol{u})) \cdot (\mathbf{a}(\boldsymbol{v}) - \mathbf{a}(\boldsymbol{u}))^2$$

Joint Optimization

Shading Constraint (data term)

- Idea: **maximize consistency** between **estimated voxel shading** and **sampled intensities** in input luminance images (gradient for robustness)

$$E_g(\boldsymbol{v}) = \sum_{\mathcal{I}_i \in \mathcal{V}_{\text{best}}} w_i^{\boldsymbol{v}} \|\nabla \mathbf{B}(\boldsymbol{v}) - \nabla \mathcal{I}_i(\pi(v_i))\|_2^2$$

Joint Optimization

Shading Constraint (data term)

- Idea: **maximize consistency** between **estimated voxel shading** and **sampled intensities** in input luminance images (gradient for robustness)

$$E_g(\boldsymbol{v}) = \sum_{\mathcal{I}_i \in \mathcal{V}_{\text{best}}} w_i^{\boldsymbol{v}} \|\nabla \mathbf{B}(\boldsymbol{v}) - \nabla \mathcal{I}_i(\pi(v_i))\|_2^2$$

Best views for voxel and respective view-dependent weights

Joint Optimization

Shading Constraint (data term)

- Idea: **maximize consistency** between **estimated voxel shading** and **sampled intensities** in input luminance images (gradient for robustness)

$$E_g(\boldsymbol{v}) = \sum_{\mathcal{I}_i \in \mathcal{V}_{\text{best}}} w_i^{\boldsymbol{v}} \|\nabla \mathbf{B}(\boldsymbol{v}) - \nabla \mathcal{I}_i(\pi(v_i))\|_2^2$$

Best views for voxel and respective view-dependent weights

Shading: allows for optimization of surface (through normal) and albedo

Joint Optimization

Shading Constraint (data term)

- Idea: **maximize consistency** between **estimated voxel shading** and **sampled intensities** in input luminance images (gradient for robustness)

$$E_g(\boldsymbol{v}) = \sum_{\mathcal{I}_i \in \mathcal{V}_{\text{best}}} w_i^{\boldsymbol{v}} \|\nabla \mathbf{B}(\boldsymbol{v}) - \nabla \mathcal{I}_i(\pi(v_i))\|_2^2$$

Best views for voxel and respective view-dependent weights

Shading: allows for optimization of surface (through normal) and albedo

Voxel center transformed and projected into input view

Joint Optimization

Shading Constraint (data term)

- Idea: **maximize consistency** between **estimated voxel shading** and **sampled intensities** in input luminance images (gradient for robustness)

$$E_g(\boldsymbol{v}) = \sum_{\mathcal{I}_i \in \mathcal{V}_{\text{best}}} w_i^{\boldsymbol{v}} \|\nabla \mathbf{B}(\boldsymbol{v}) - \nabla \mathcal{I}_i(\pi(v_i))\|_2^2$$

Best views for voxel and respective view-dependent weights

Shading: allows for optimization of surface (through normal) and albedo

Voxel center transformed and projected into input view

Sampling: allows for optimization of camera poses and camera intrinsics

Recolorization

Optimal colors

- **Recompute voxel colors** after optimization at each level

Recolorization

Optimal colors

- **Recompute voxel colors** after optimization at each level
- Sampling
  - Sample from **keyframes only**
  - **Collect, weight and filter** observations

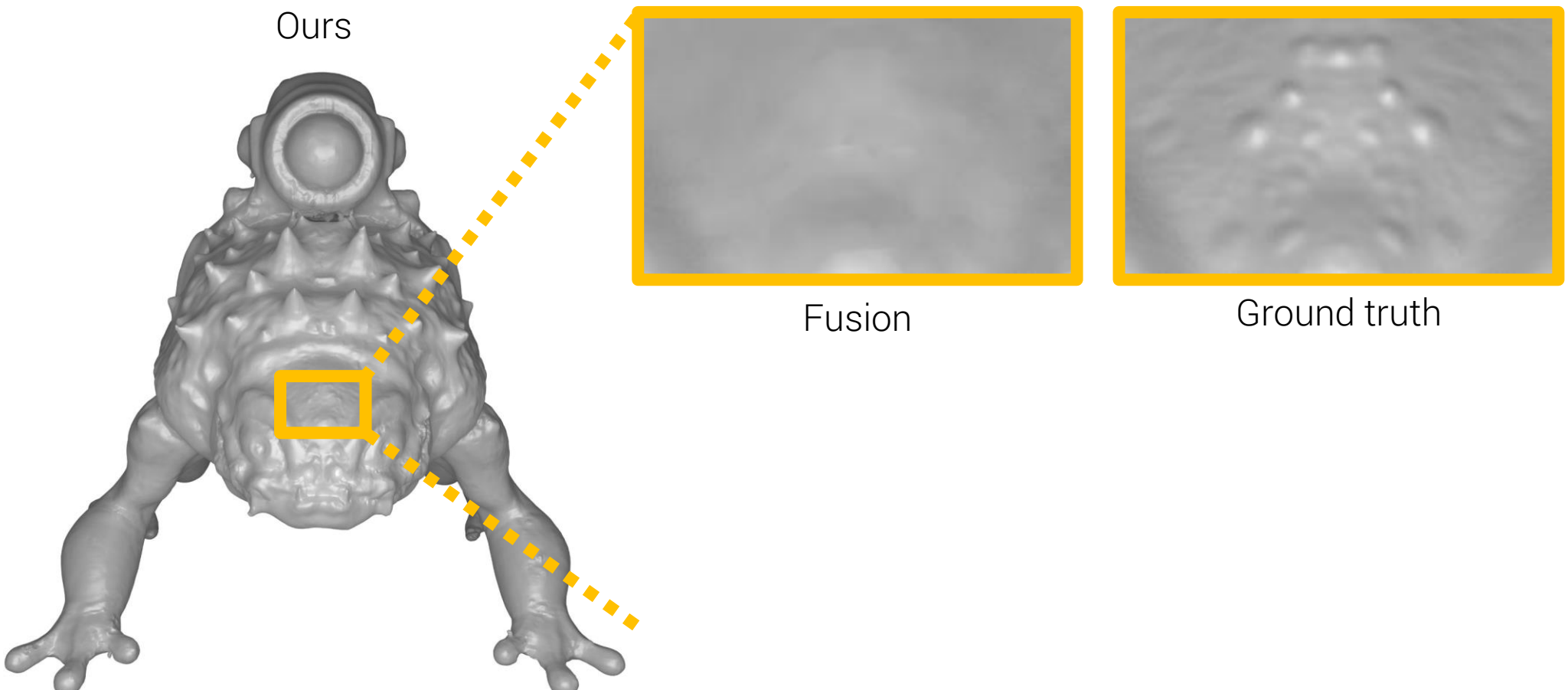
Recolorization

Optimal colors

- **Recompute voxel colors** after optimization at each level
- Sampling
  - Sample from **keyframes only**
  - **Collect, weight and filter** observations
- **Weighted average** $c_{\boldsymbol{v}}^* = \arg\min_{c_{\boldsymbol{v}}} \sum_{(c_i^{\boldsymbol{v}}, w_i^{\boldsymbol{v}}) \in \mathcal{O}_v} w_i^{\boldsymbol{v}} (c_{\boldsymbol{v}} - c_i^{\boldsymbol{v}})^2.$

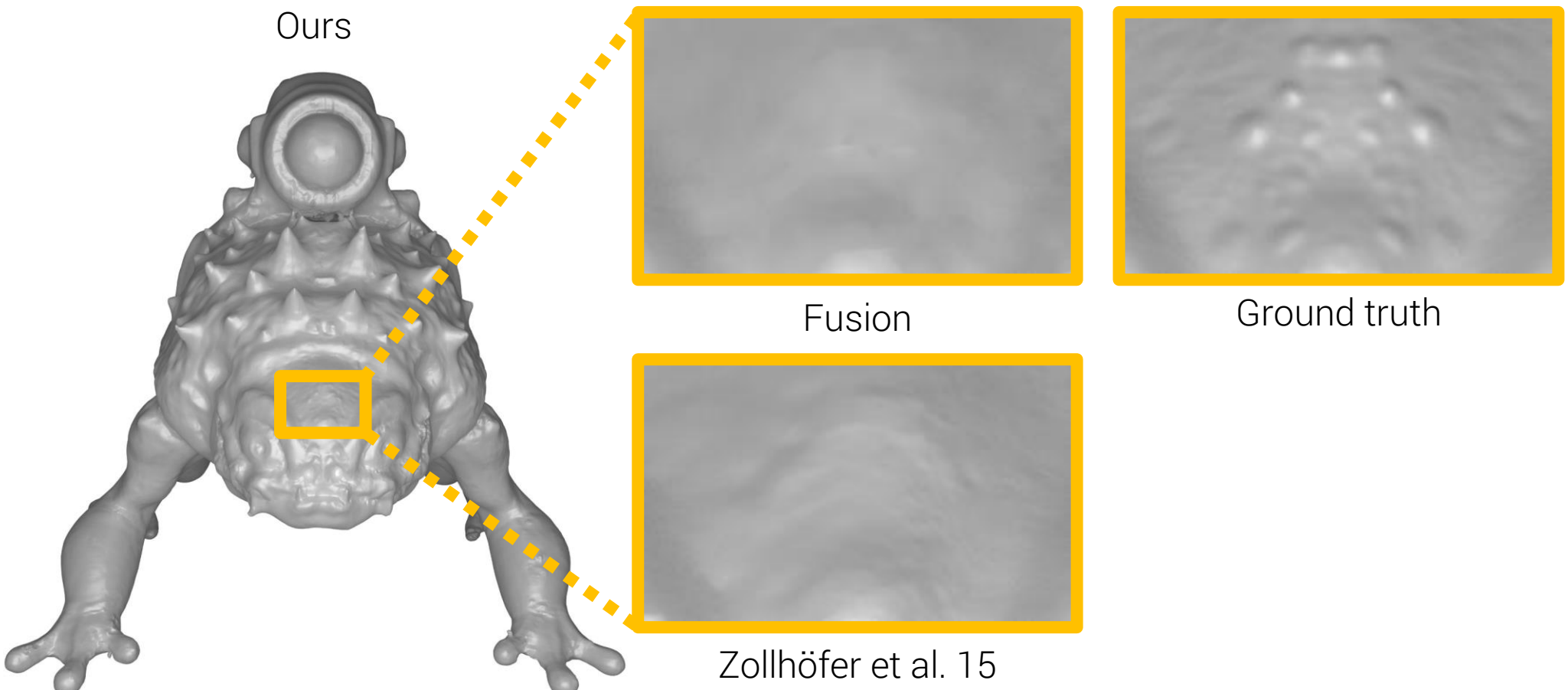
Results: Ground Truth Geometry

Frog (synthetic)



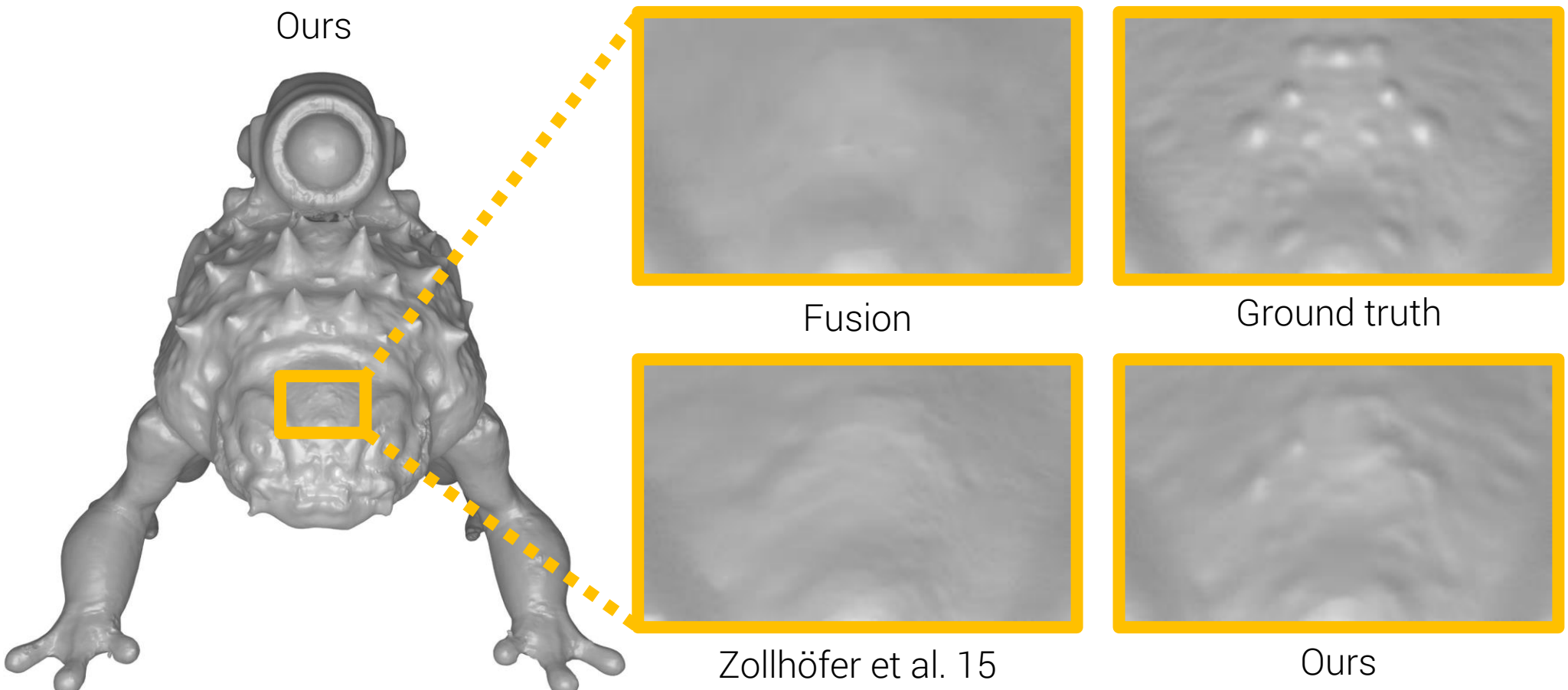
Results: Ground Truth Geometry

Frog (synthetic)



Results: Ground Truth Geometry

Frog (synthetic)

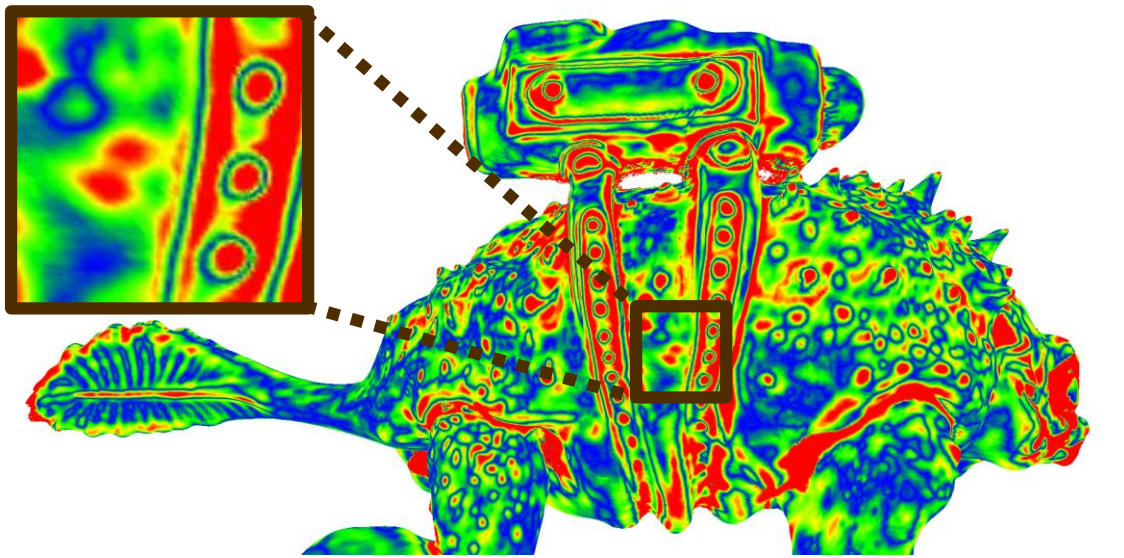


Results: Ground Truth Geometry

Frog (synthetic)

- Quantitative surface accuracy evaluation
- Color coding: absolute distances (ground truth)

Zollhöfer et al. 15

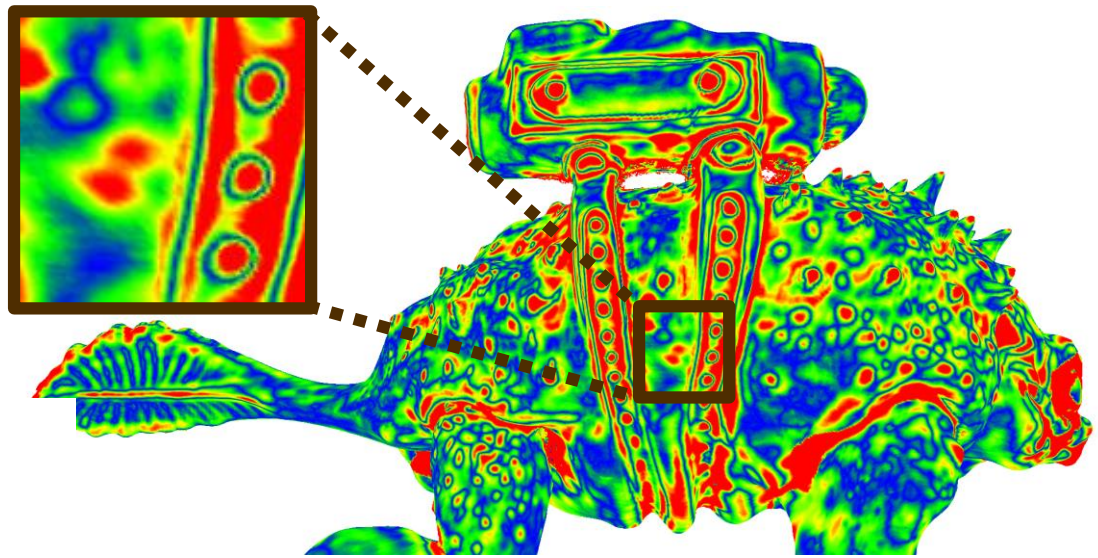


Results: Ground Truth Geometry

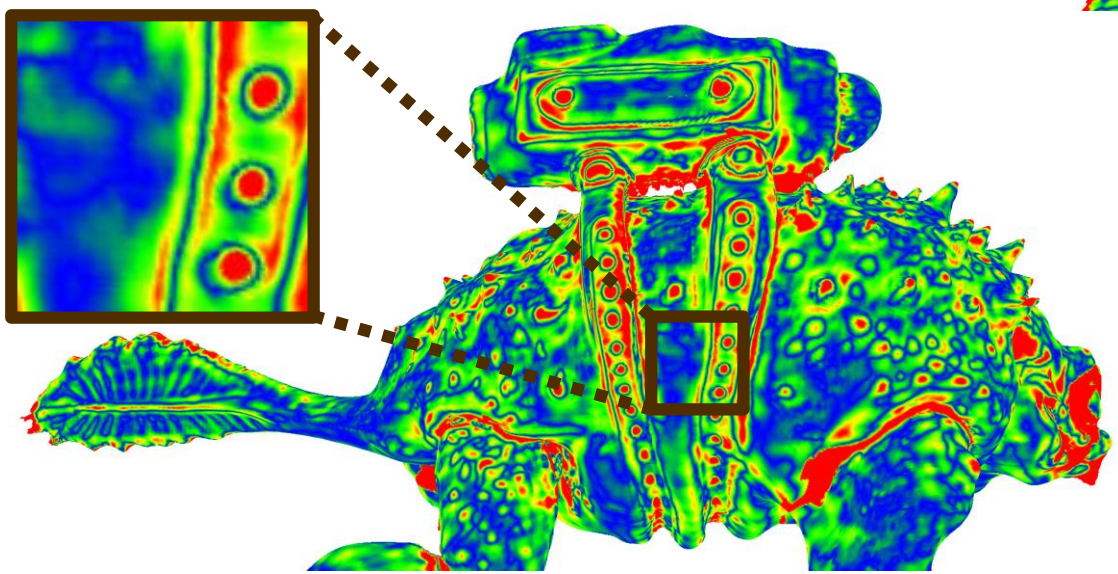
Frog (synthetic)

- Quantitative surface accuracy evaluation
- Color coding: absolute distances (ground truth)

Zollhöfer et al. 15



Ours

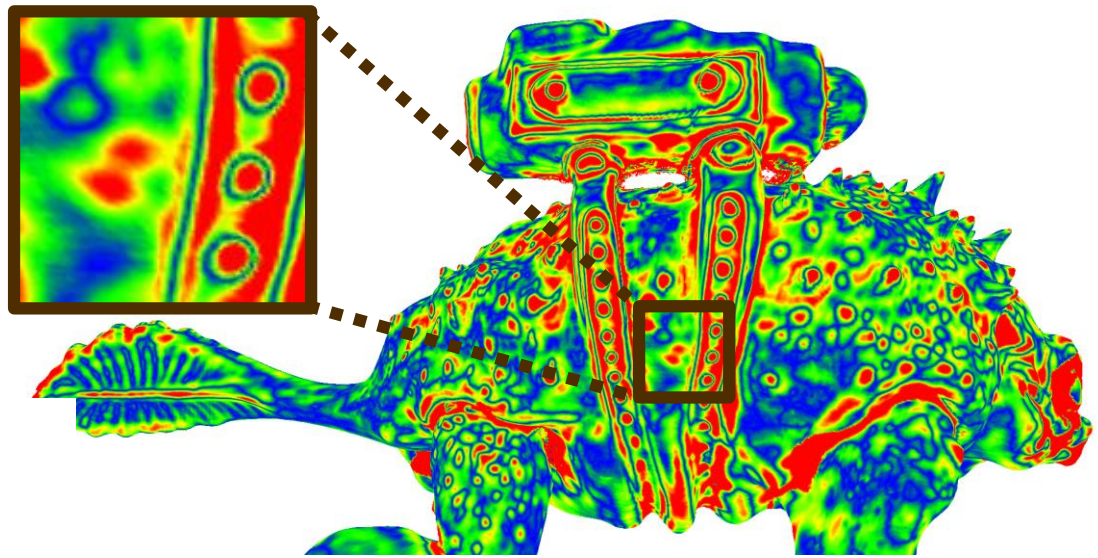


Results: Ground Truth Geometry

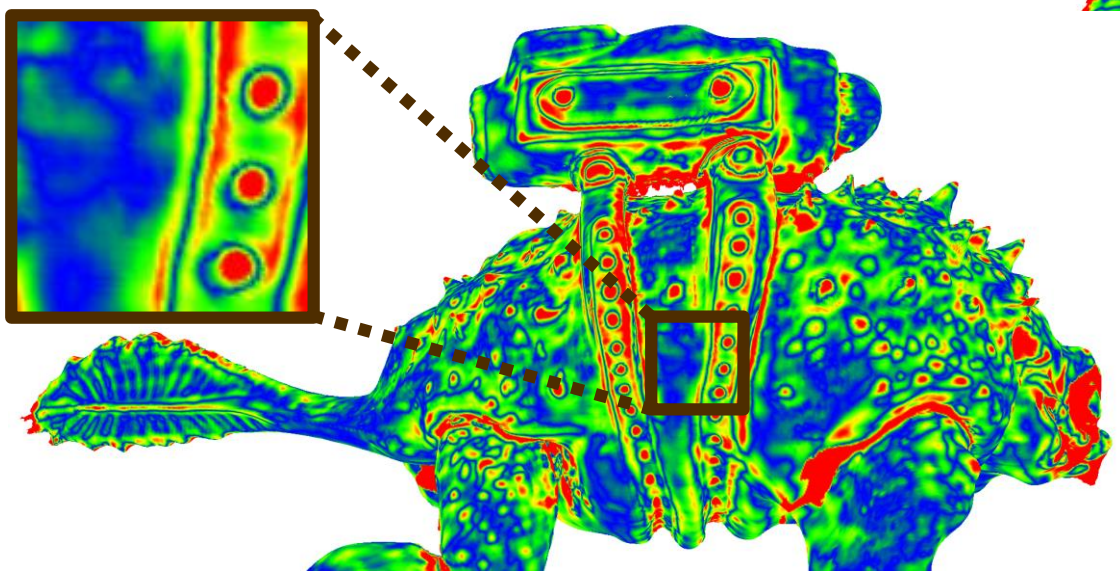
Frog (synthetic)

- Quantitative surface accuracy evaluation
- Color coding: absolute distances (ground truth)

Zollhöfer et al. 15



Ours



Mean absolute deviation:

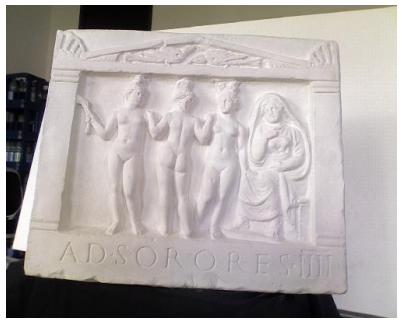
- Ours: 0.222mm (std.dev. 0.269mm)
- Zollhöfer et al: 0.278mm (std.dev. 0.299mm)

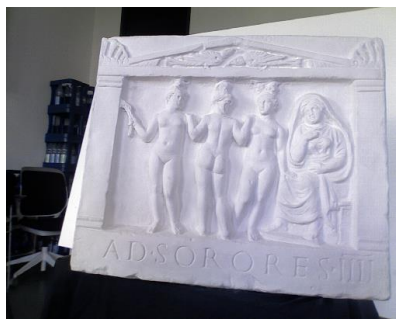
  → 20.14% more accurate

Qualitative Results

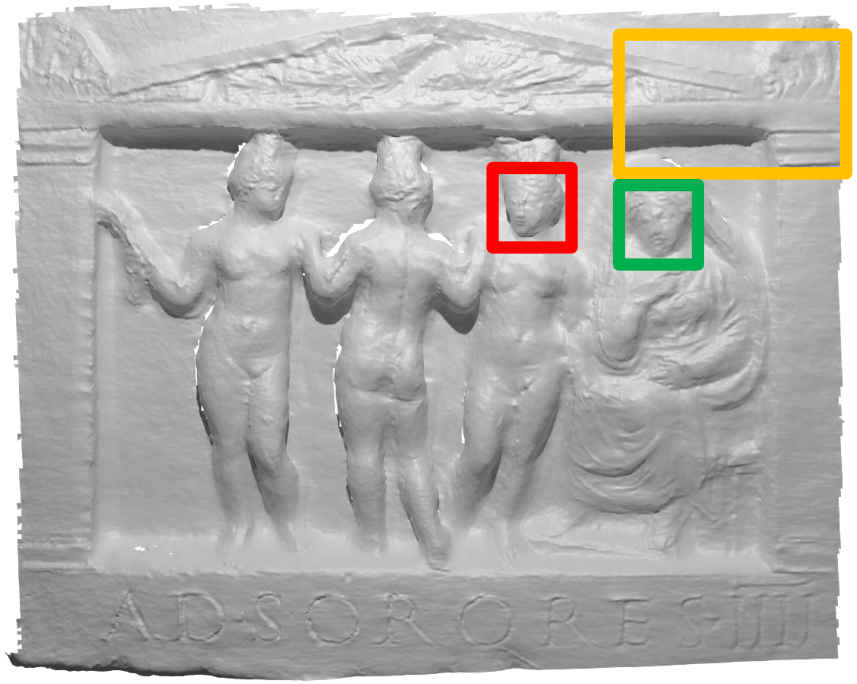
Relief (geometry)

Input Color

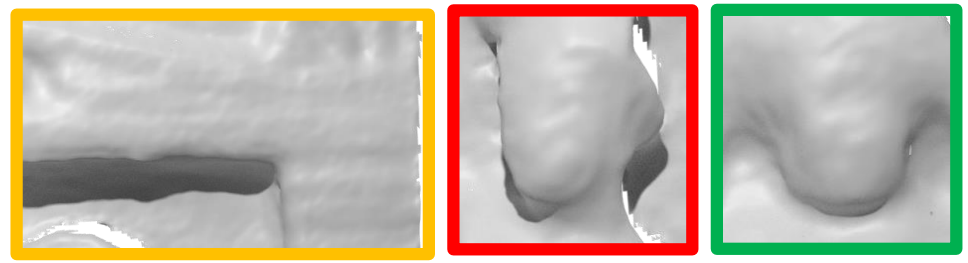




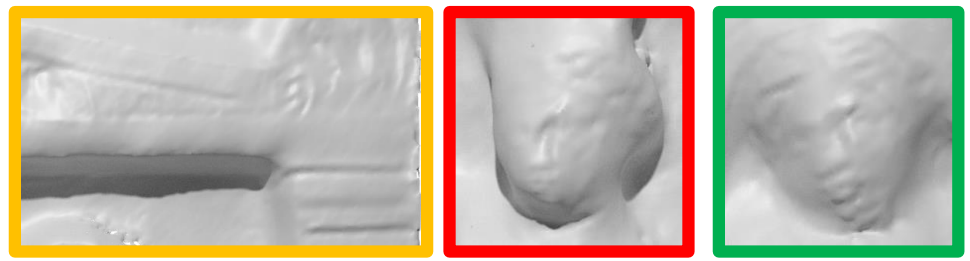
Ours



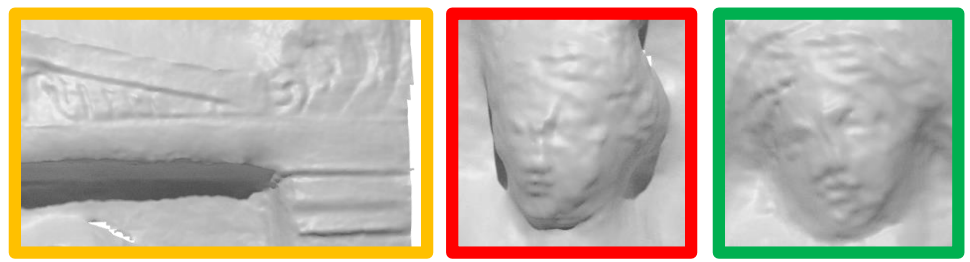
Fusion



Zollhöfer et al. 15



Ours



Qualitative Results

Fountain (appearance)

Input Color

Ours

Fusion

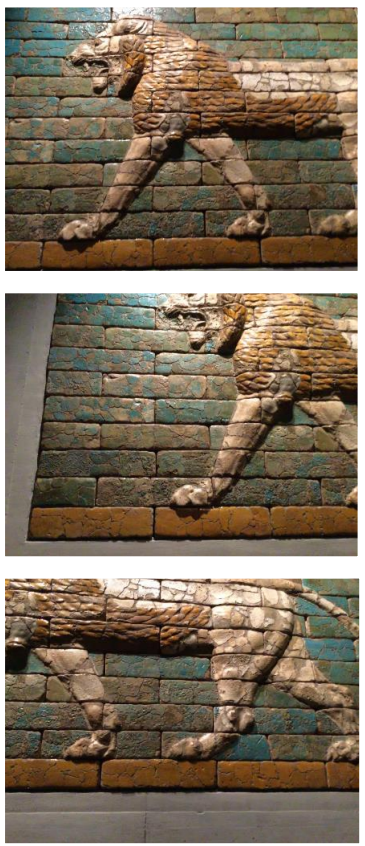
Zollhöfer et al. 15

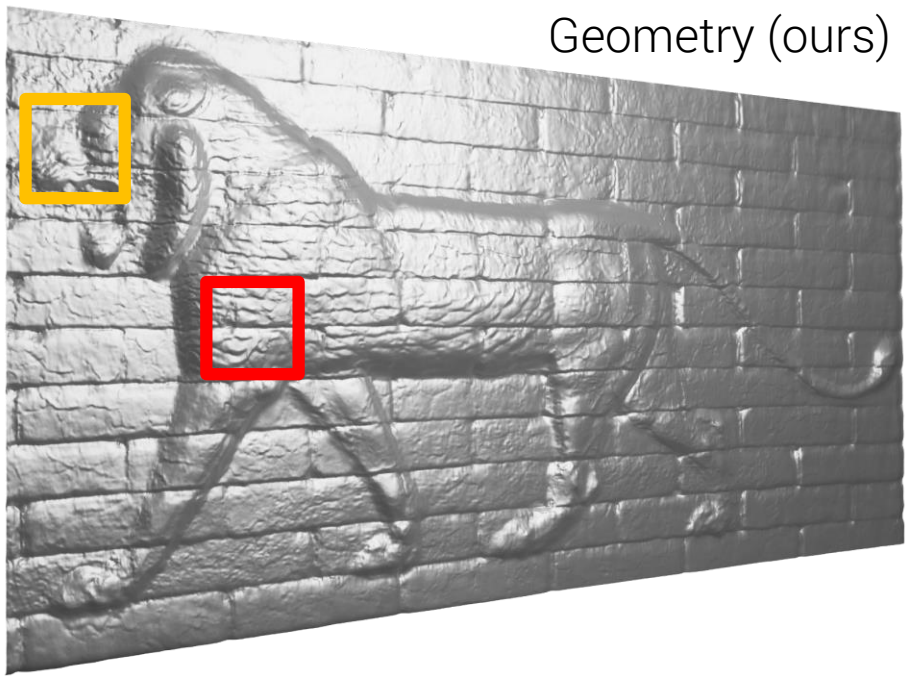
Ours

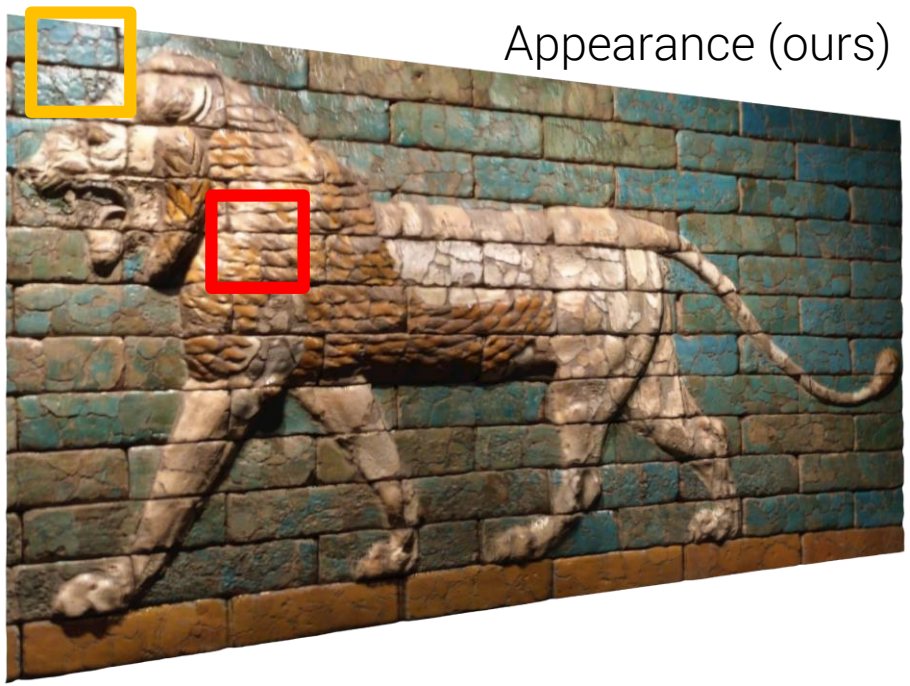
Qualitative Results

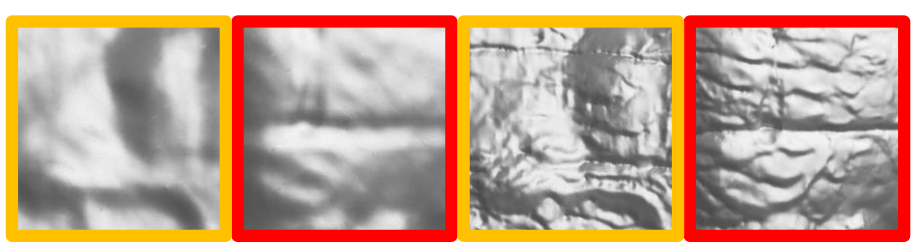
Lion

Input Color

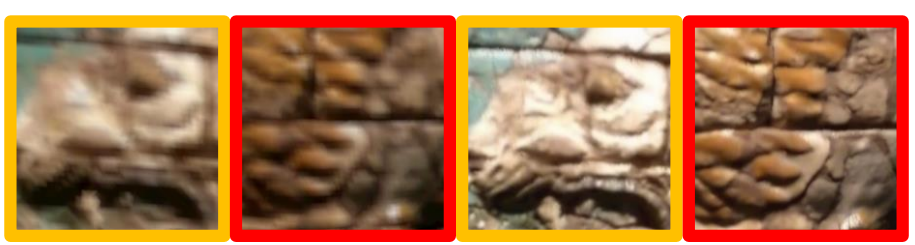








Fusion Ours

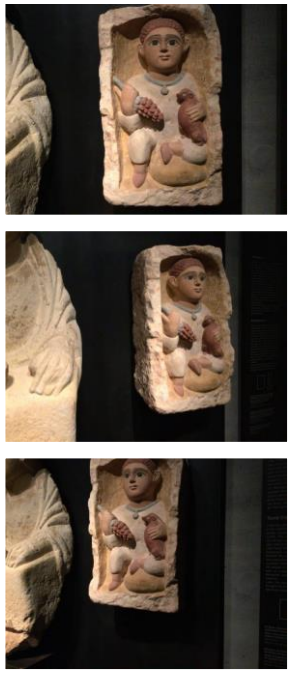


Fusion Ours

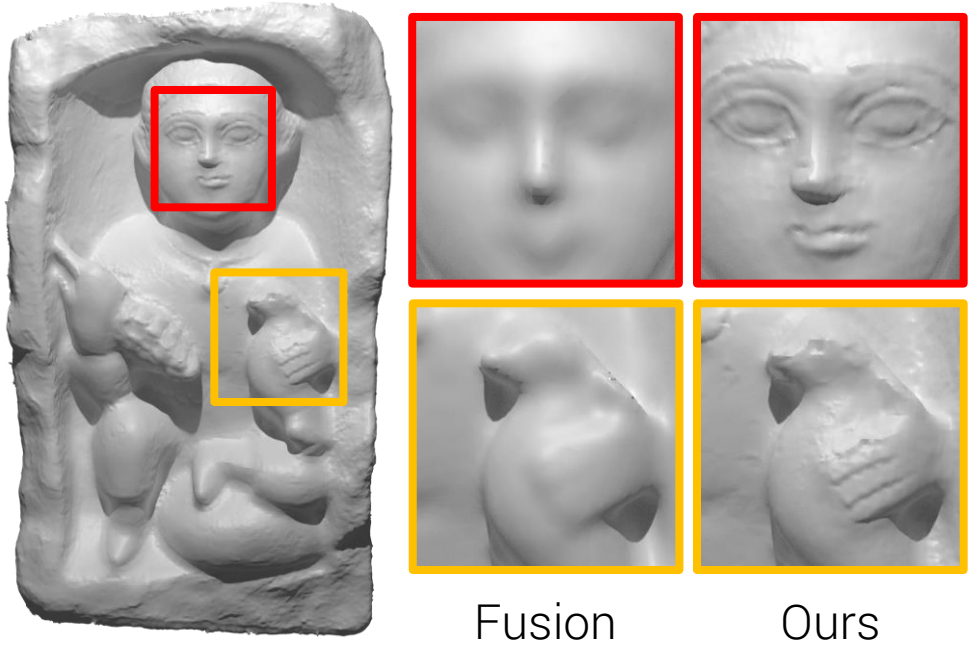
Qualitative Results

Tomb Statuary

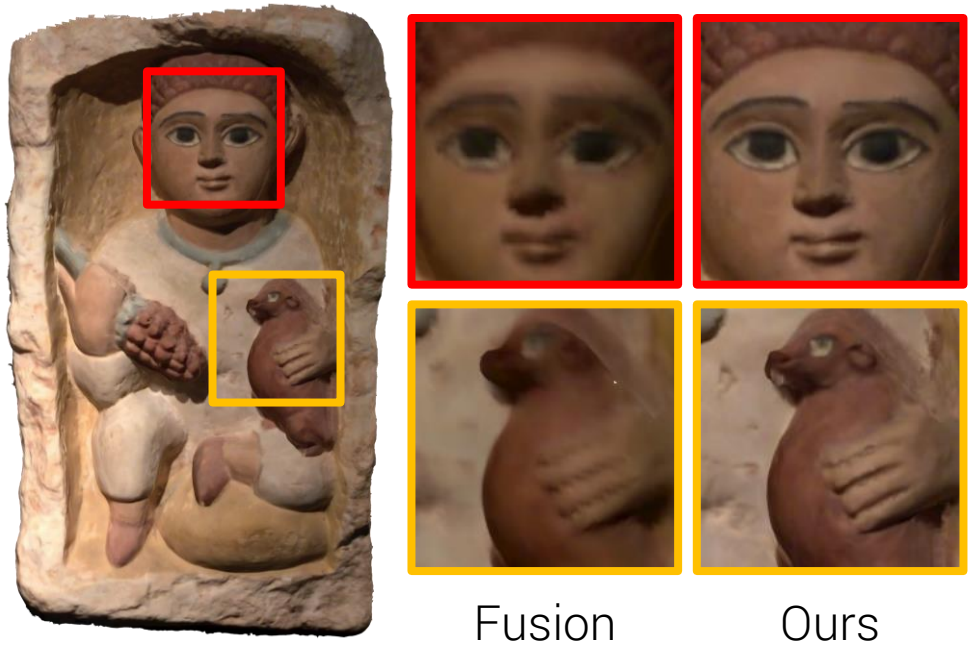
Input Color



Geometry (ours)



Appearance (ours)



Shading: Global SH vs. SVSH

Fountain

Luminance

Shading: Global SH vs. SVSH

Fountain

Luminance

Albedo

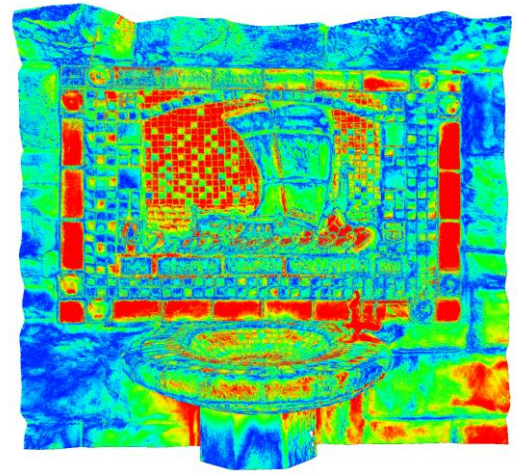
Shading: Global SH vs. SVSH

Fountain

$$\mathbf{B}_{\text{diff}} = |\mathbf{B}(\boldsymbol{v}) - \mathbf{I}(\boldsymbol{v})|$$

Luminance

Shading



Difference

Global SH

Albedo

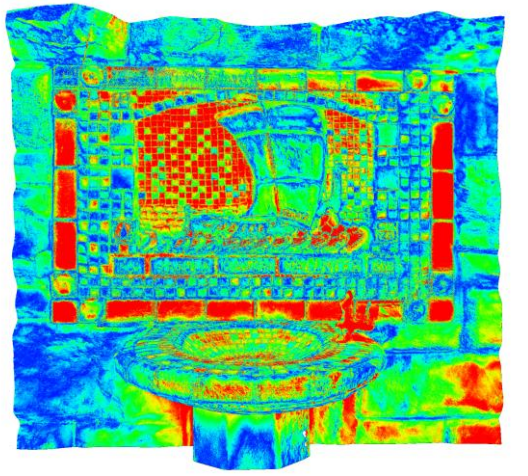
Shading: Global SH vs. SVSH

Fountain

$$\mathbf{B}_{\text{diff}} = |\mathbf{B}(\boldsymbol{v}) - \mathbf{I}(\boldsymbol{v})|$$

Luminance

Shading

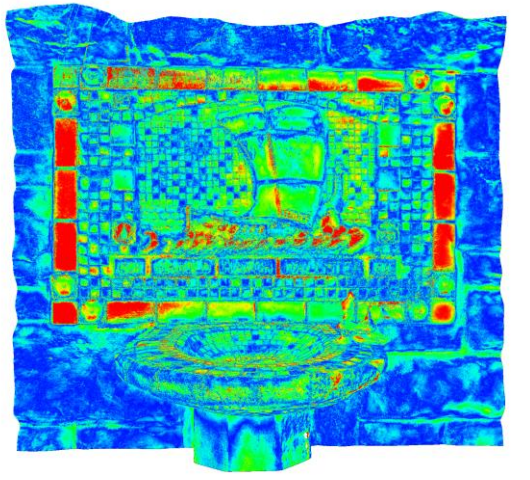


Difference

Global SH

Albedo

Shading



Difference

SVSH

Conclusion

High-Quality 3D Reconstruction of Geometry and Appearance

- Temporal view **sampling & filtering** techniques
- **Spatially-Varying Lighting** estimation
- **Joint optimization** of **surface & albedo** (SDF) and **image formation model**
- **Optimized texture** as by-product

Conclusion

High-Quality 3D Reconstruction of Geometry and Appearance

- Temporal view **sampling & filtering** techniques
- **Spatially-Varying Lighting** estimation
- **Joint optimization** of **surface & albedo** (SDF) and **image formation model**
- **Optimized texture** as by-product

Thank you!

Questions?

Robert Maier

Technical University of Munich
Computer Vision Group

robert.maier@in.tum.de
https://vision.in.tum.de/members/maierr