Efficient Plane-Based Optimization of Geometry and Texture for Indoor RGB-D Reconstruction

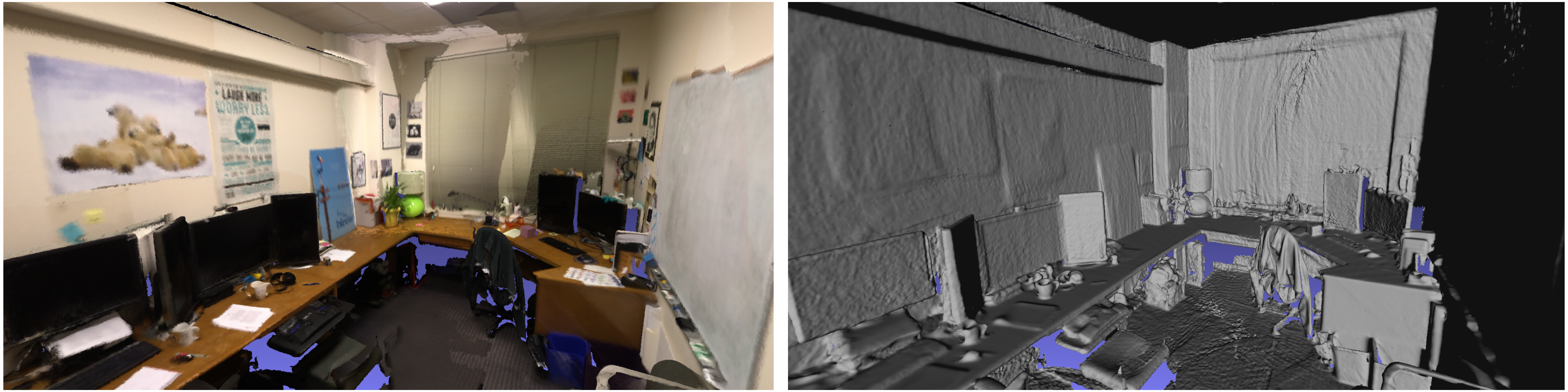
Chao Wang and Xiaohu Guo

University of Texas at Dallas

CVPR
LONG BEACH
CALIFORNIA
June 16-20, 2019

Models from online 3D reconstruction

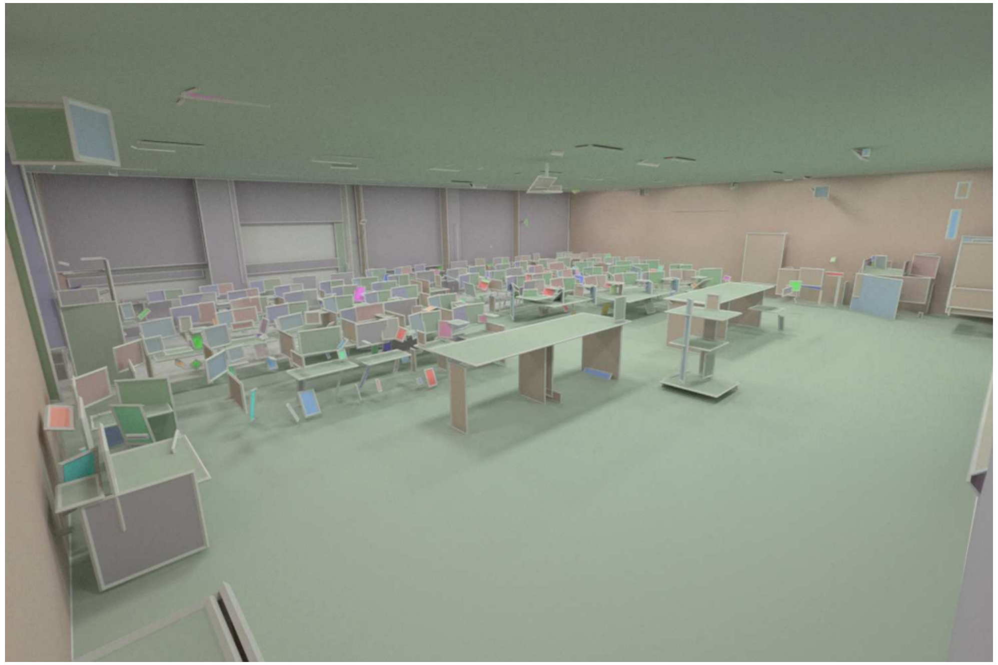
- Dense and noisy model with blurry textures, artifacts, misalignment



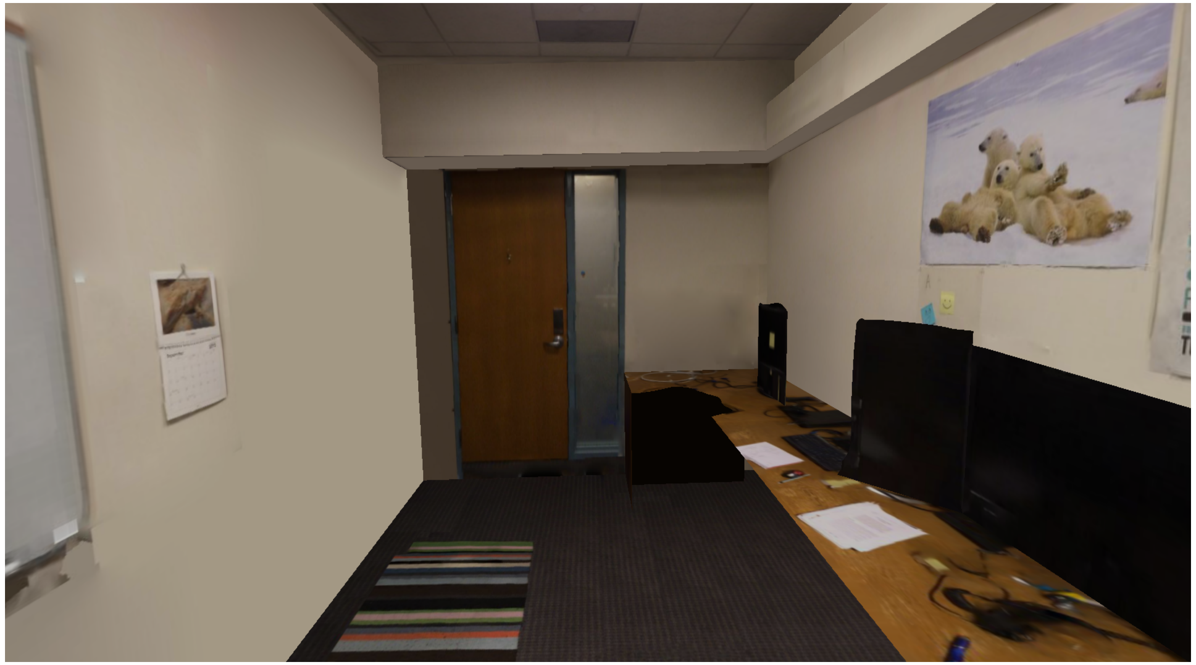
office0 from BundleFusion dataset (Dai et al., TOG’17)
2.9M vertices, 5.6M faces

Current plane-based optimization methods

- Work on building framework only or large planar areas only



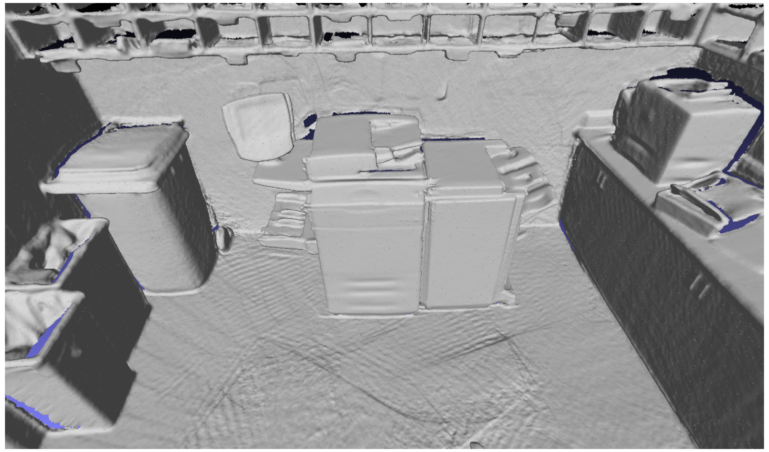
RAPTER (Monszpart et al., Siggraph’15)



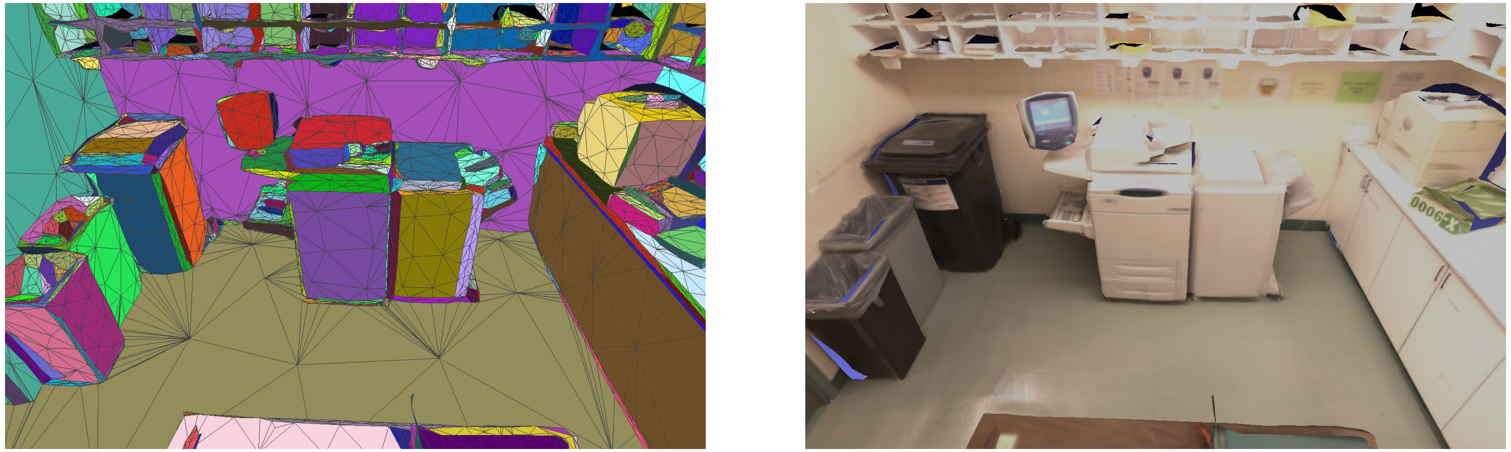
3DLite (Huang et al., TOG’17)

Our method

- **Input**: RGB-D sequence and dense mesh reconstructed from it
- **Output**: lightweight, low-polygon mesh with textures



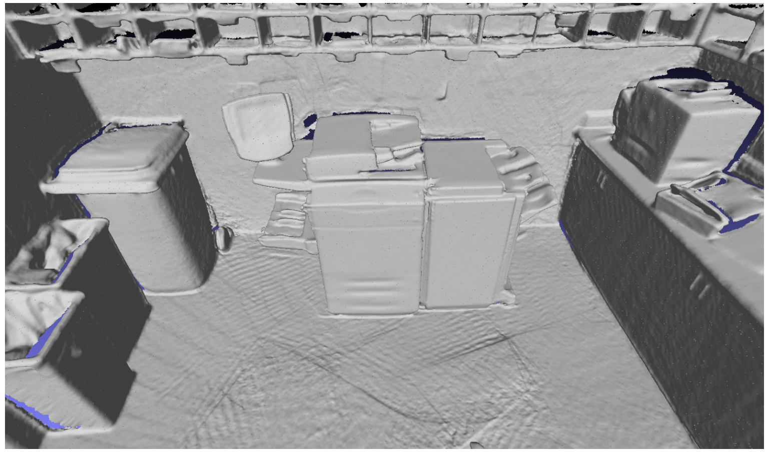
Input dense model by BundleFusion
3.70M vertices, 7.28M faces



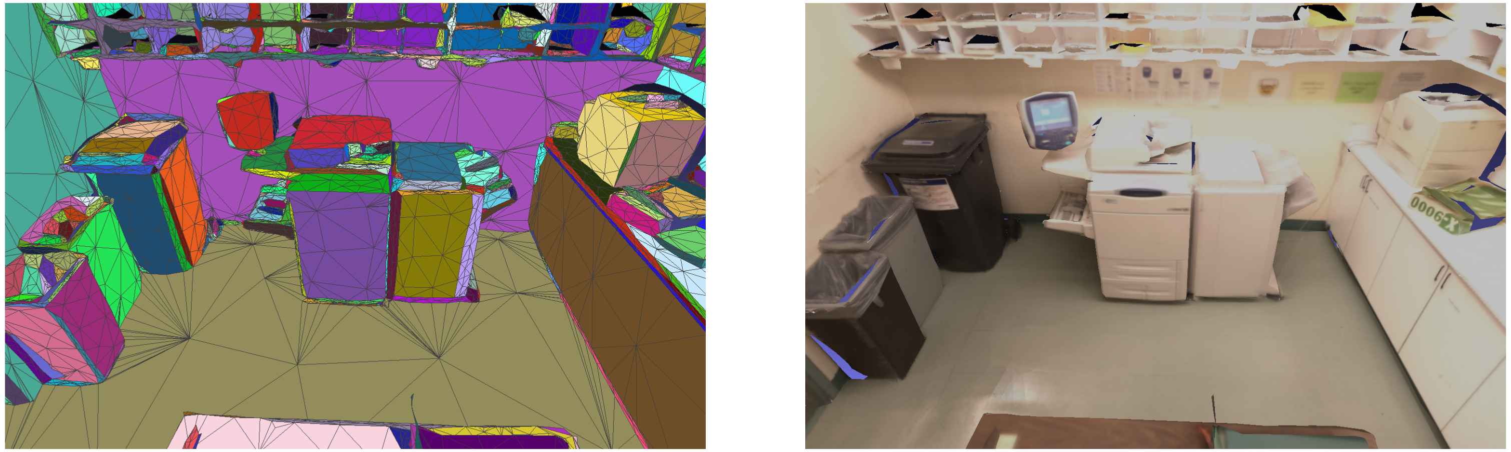
Output plane partition and textured mesh
16K vertices, 31K faces

Pros

- Build entire scene by planes without losing details;
- Preserve sharp features;
- Efficient: 10-20 minutes per model instead of hours in state-of-the-arts on same sequences.



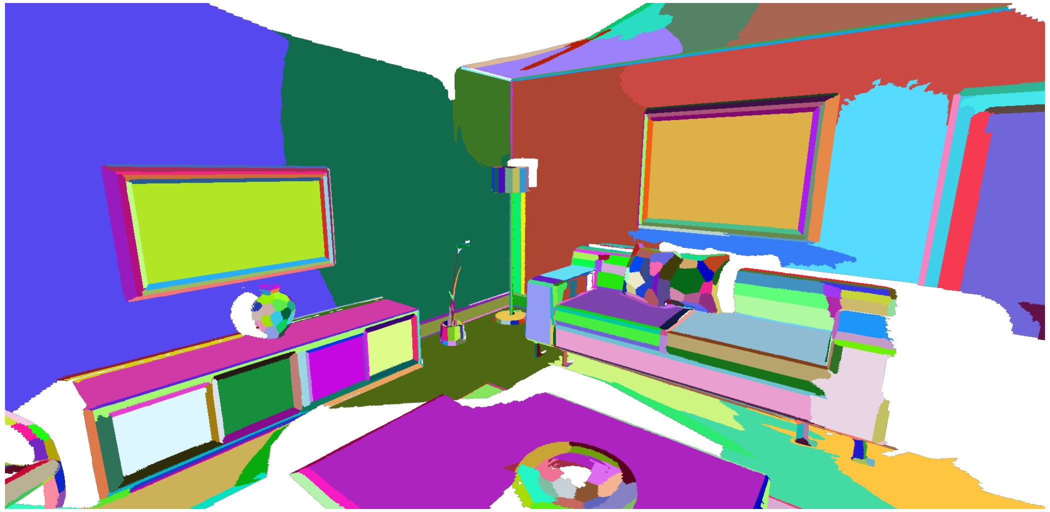
Input dense model by BundleFusion
3.70M vertices, 7.28M faces



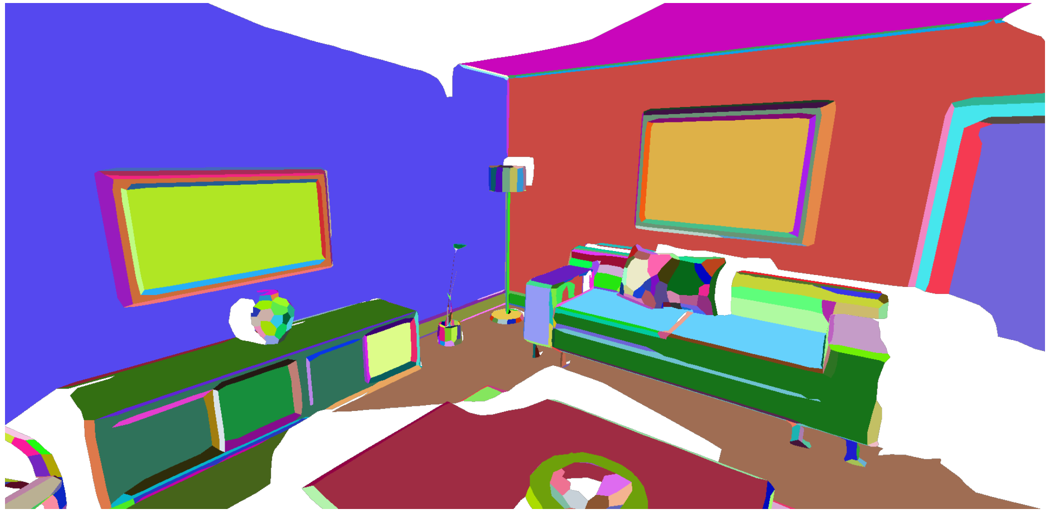
Output plane partition and textured mesh
16K vertices, 31K faces

1. Planar partition

- PCA-energy-based surface partition algorithm (Cai et al., TVCG’17)
- Merge neighbor planes



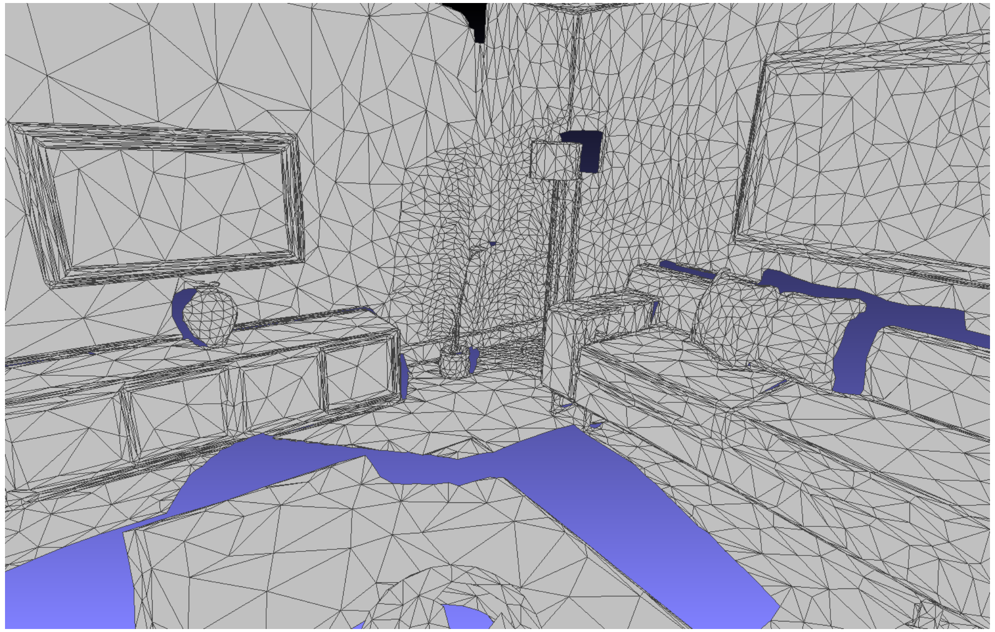
Initial planes



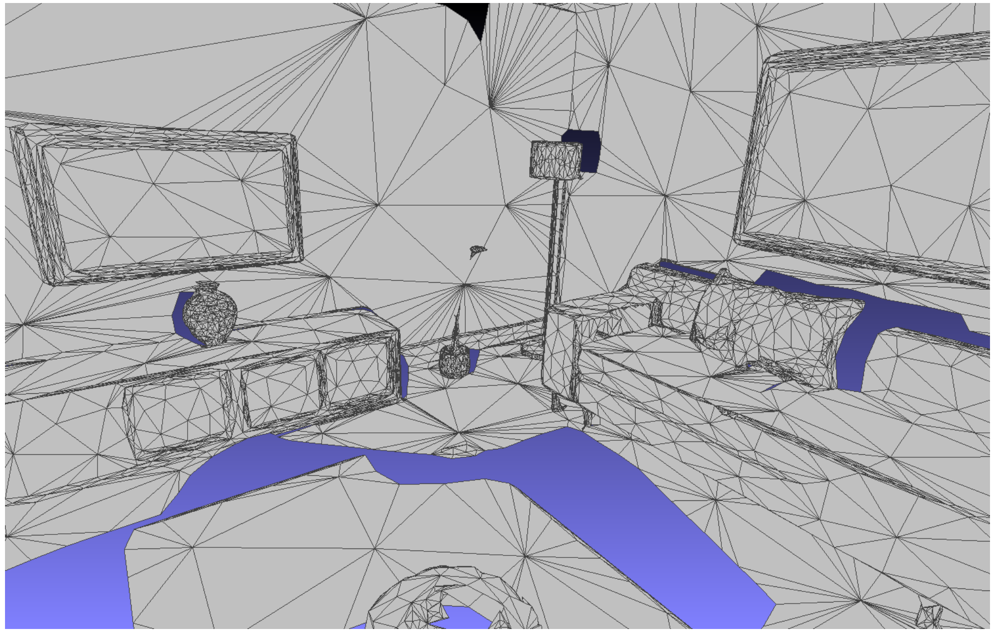
Refined planes

2. Mesh simplification based on planes

- Use quadric error metric (QEM)
  - Simplify inner plane areas first
  - Simplify all plane borders next



Common global QEM



Ours

3. Plane, texture and pose optimization

$$E_{tex}(\mathbf{T},\boldsymbol{\Phi},\mathbf{C}) = \underbrace{E_c(\mathbf{T},\boldsymbol{\Phi},\mathbf{C})}_{\boxed{\text{Photometric consistency}}} + \lambda_p \underbrace{E_p(\boldsymbol{\Phi})}_{\boxed{\text{Plane constraint}}} + \lambda_t \underbrace{E_t(\mathbf{T},\boldsymbol{\Phi})}_{\boxed{\text{Line constraint}}}$$

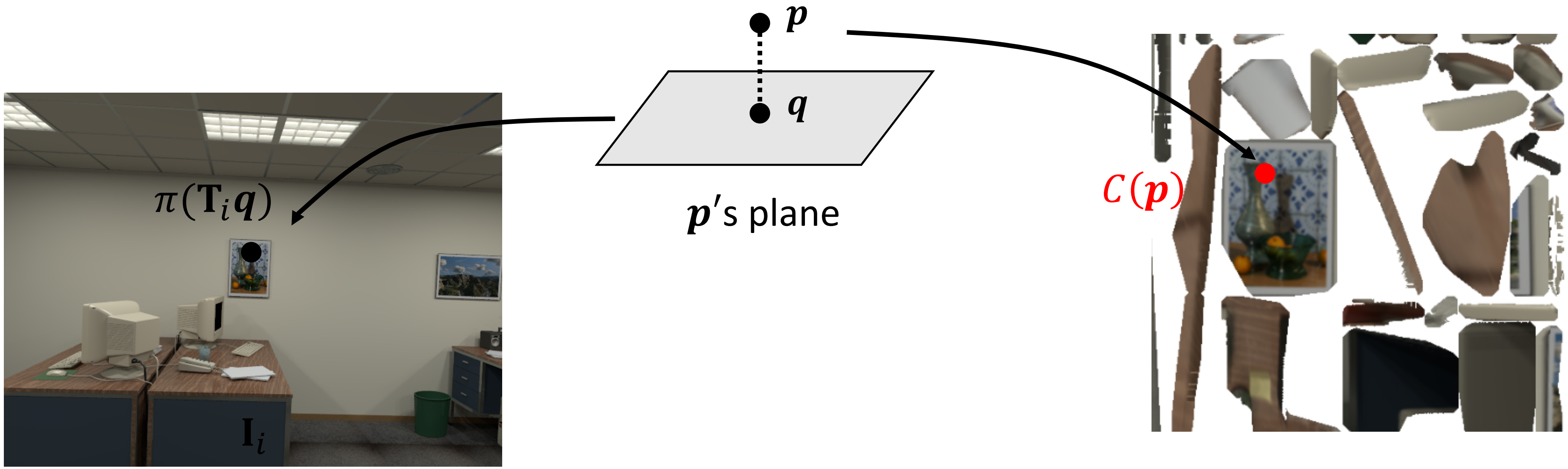
$\mathbf{T}$: camera poses, each with 6DoF

$\boldsymbol{\Phi}$: plane parameters: 4DoF

$\mathbf{C}$: texture image pixel (texel) colors

Photometric consistency term

$$E_c(\mathbf{T},\boldsymbol{\Phi},\mathbf{C},\mathbf{F}) = \sum_{\boldsymbol{i}} \sum_{\boldsymbol{p}} ||C(\boldsymbol{p}) - \mathbf{I}_i(\pi(\mathbf{T}_i\boldsymbol{q}))||^2$$

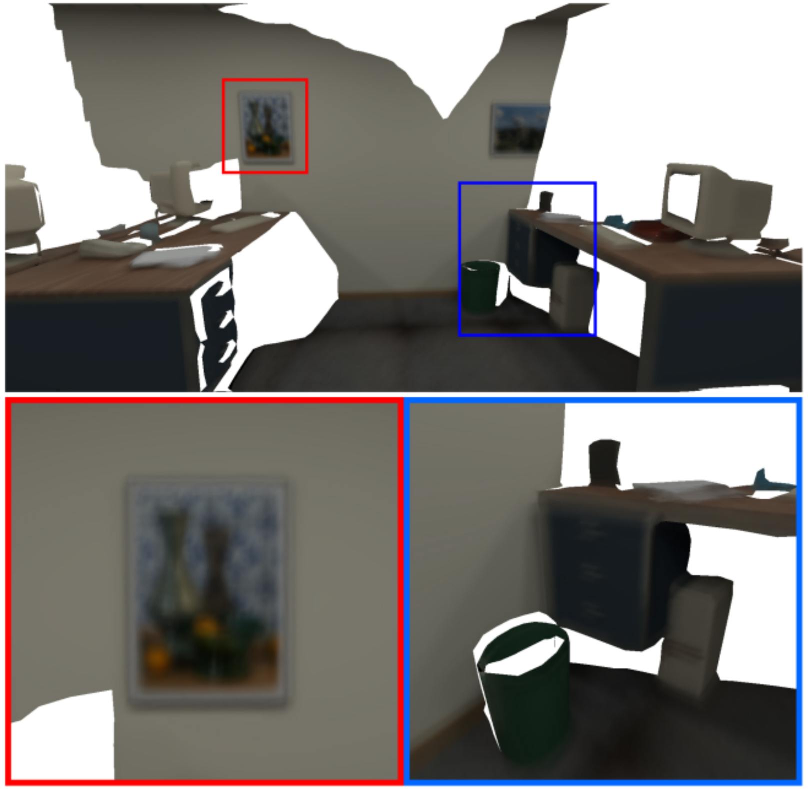


Color image in frame i

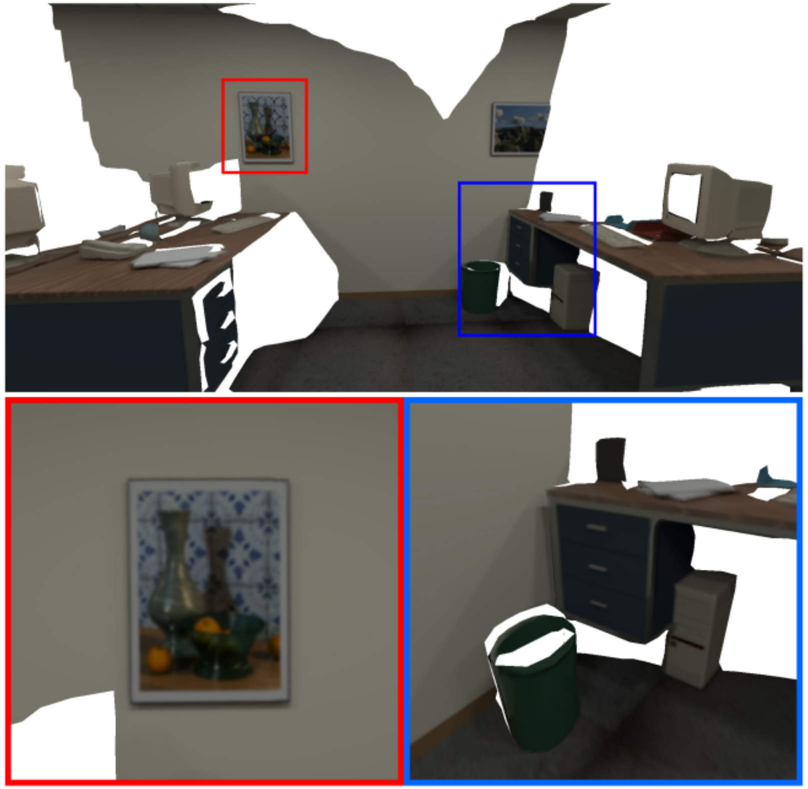
World space

Texture image

Optimization result



No optimization



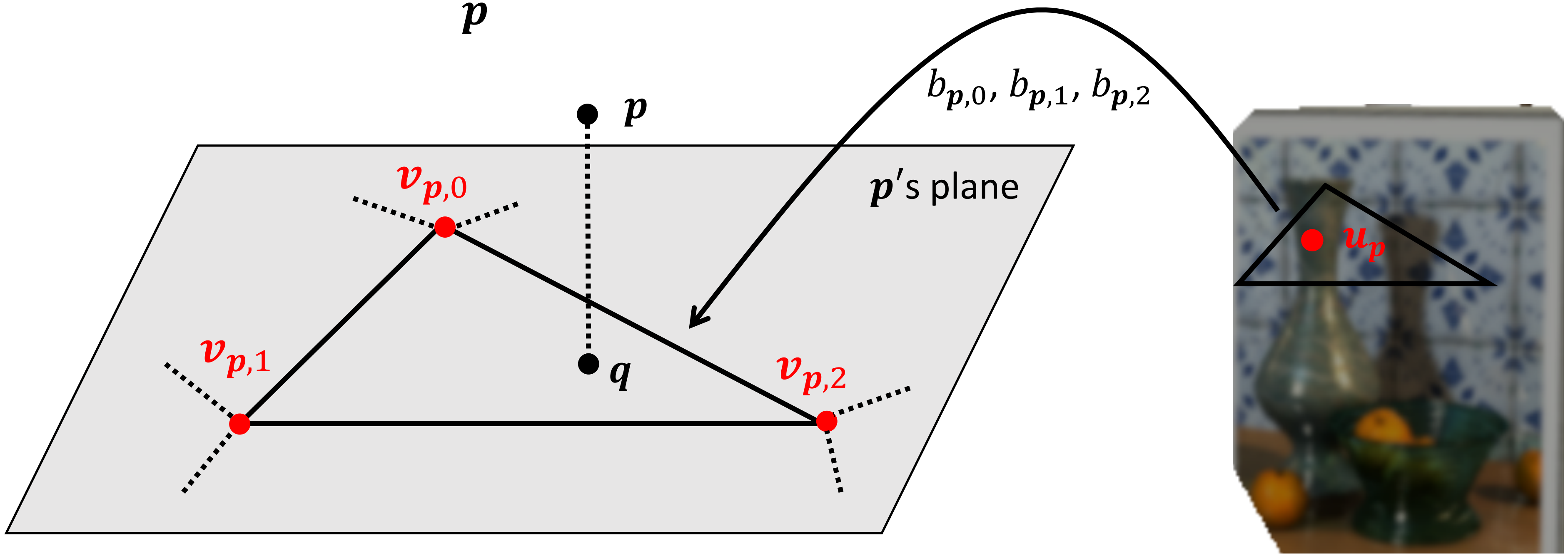
With optimization

4. Geometry optimization

$$E_{vert}(\mathbf{V}) = \underbrace{E_g(\mathbf{V})}_{\boxed{\text{Vetex-plane consistency}}} + \lambda_l \underbrace{E_l(\mathbf{V})}_{\boxed{\text{Line constraint}}} + \lambda_r \underbrace{E_r(\mathbf{V})}_{\boxed{\text{Regularization based on neighbor connectivity}}}$$

Vertex-plane consistency

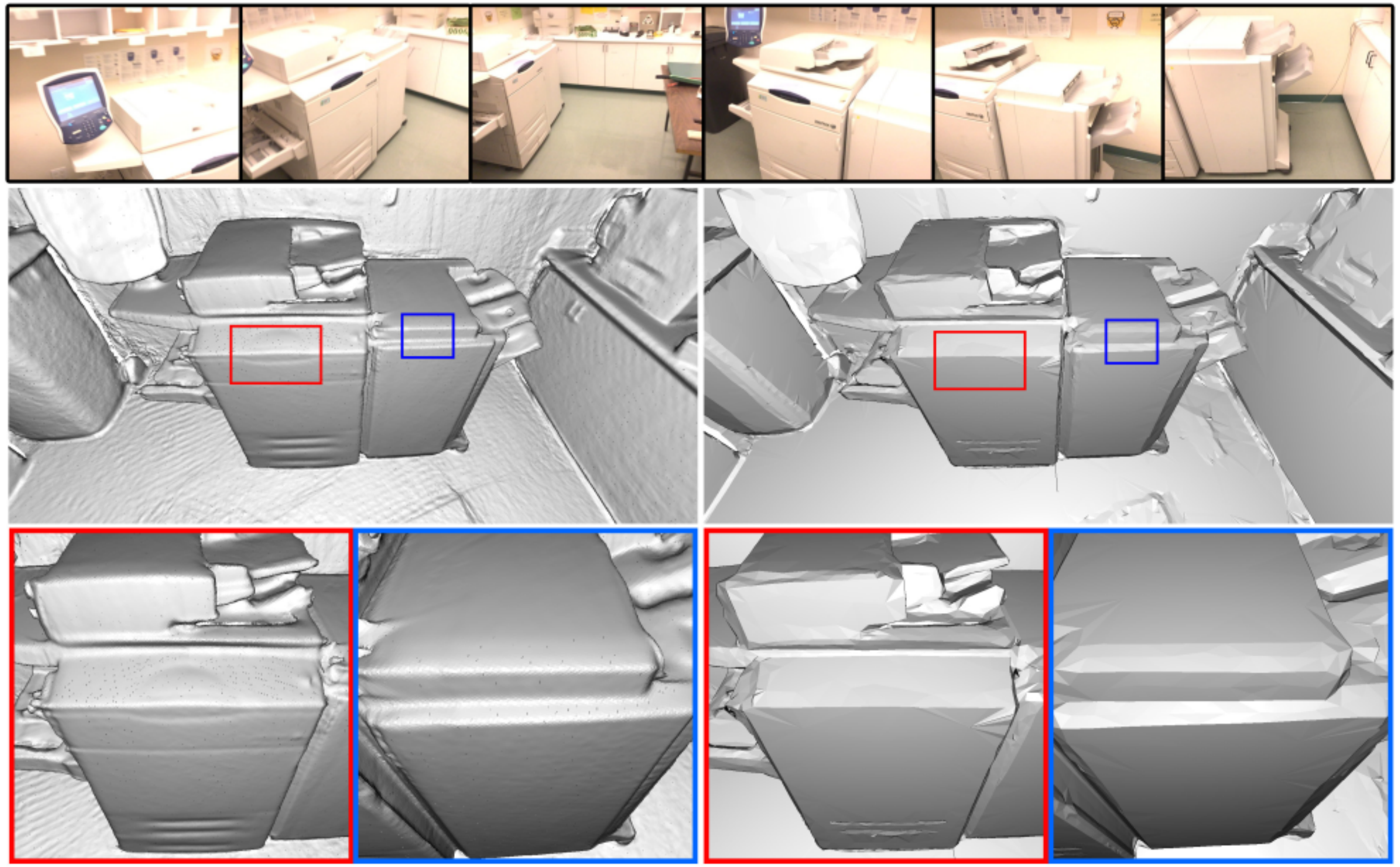
$$E_g(\mathbf{V}) = \sum_{\boldsymbol{p}} ||\boldsymbol{q} - (b_{\boldsymbol{p},0}\boldsymbol{v}_{\boldsymbol{p},0} + b_{\boldsymbol{p},1}\boldsymbol{v}_{\boldsymbol{p},1} + b_{\boldsymbol{p},2}\boldsymbol{v}_{\boldsymbol{p},2})||^2$$


$b_{\boldsymbol{p},0}, b_{\boldsymbol{p},1}, b_{\boldsymbol{p},2}$
$\boldsymbol{p}$
$\boldsymbol{p}$'s plane
$\boldsymbol{v}_{\boldsymbol{p},0}$
$\boldsymbol{v}_{\boldsymbol{p},1}$
$\boldsymbol{v}_{\boldsymbol{p},2}$
$\boldsymbol{q}$
$\boldsymbol{u}_{\boldsymbol{p}}$

World space

Texture image

$b_{\boldsymbol{p},0}, b_{\boldsymbol{p},1}, b_{\boldsymbol{p},2}$: $\boldsymbol{u}_{\boldsymbol{p}}$'s barycentric coordinates inside its triangle on texture image



Input fused dense mesh

After geometry optimization

Model: ***office0*** (from BundleFusion dataset)

3DLite
(41k vertices,
63K faces)

1x speed

BundleFusion
(5.71M vertices, 11.3M faces)

Ours
(24k vertices,
42K faces)

Thank you!

Source code can be found in
https://github.com/chaowang15/plane-opt-rgbd