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

Chao Wang, Xiaohu Guo
University of Texas at Dallas

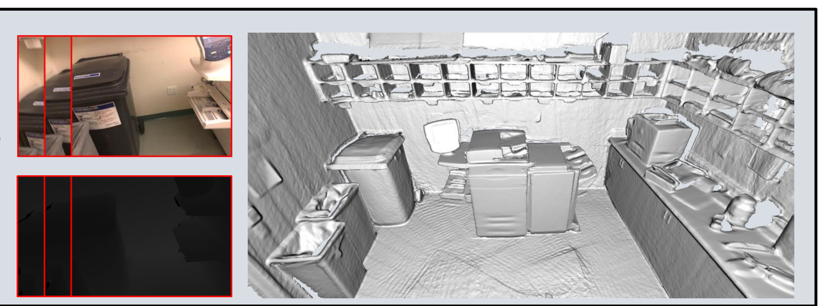
Contribution

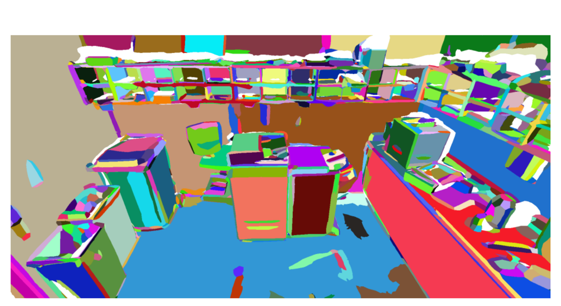
We present a novel approach to reconstruct RGB-D indoor scene with **plane primitives.** Our approach takes as input a RGB-D sequence and a dense coarse mesh reconstructed by some 3D reconstruction method on the sequence, and generate a lightweight, low-polygonal mesh with clear face textures and sharp features without losing geometry details from the original scene. Compared to existing planar reconstruction methods which only cover large planar regions in the scene, our method builds the entire scene by adaptive planes without losing geometry details and preserves **sharp features** in the final mesh.

Overview

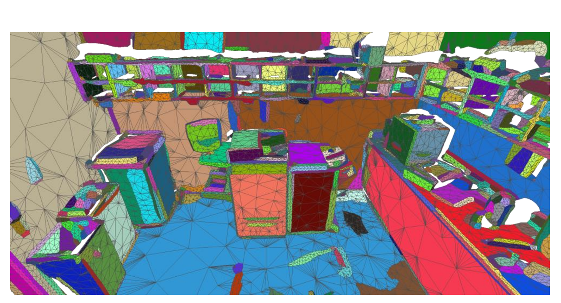
We firstly partition the input dense mesh with plane primitives, simplify it into a lightweight mesh next, then optimize plane parameters, camera poses and texture colors to maximize photometric consistency across frames, and finally optimize mesh geometry to maximize consistency between geometry and planes.

Input:
RGB-D sequence and dense mesh



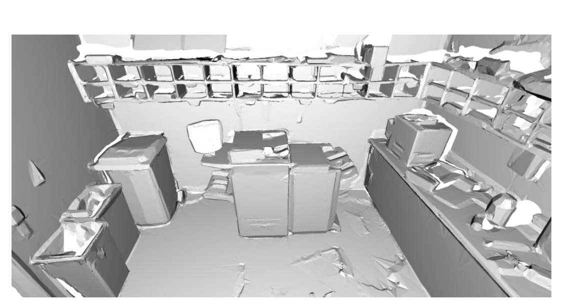


1. Planar partition



2. Mesh simplification

3. Plane, texture and pose optimization



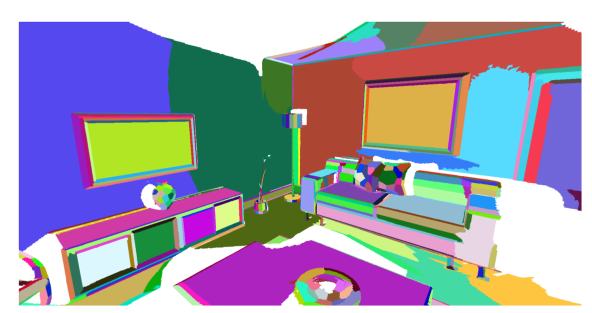
4. Geometry optimization

Model: *copyroom* from BundleFusion [TOG'17, Dai et al.]
Input mesh: 3.70M vertices, 7.28M faces
Output mesh: 55.2K vertices, 104K faces
Processing time: 1850s (CPU only)

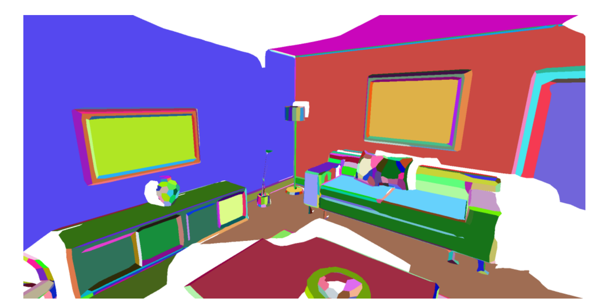
Method

1. Planar partition

Use PCA-energy-based surface partition algorithm [TVCG'17 by Cai et al.] to partition mesh surface into planes, and merge neighbor planes together if they are nearly coplanar.



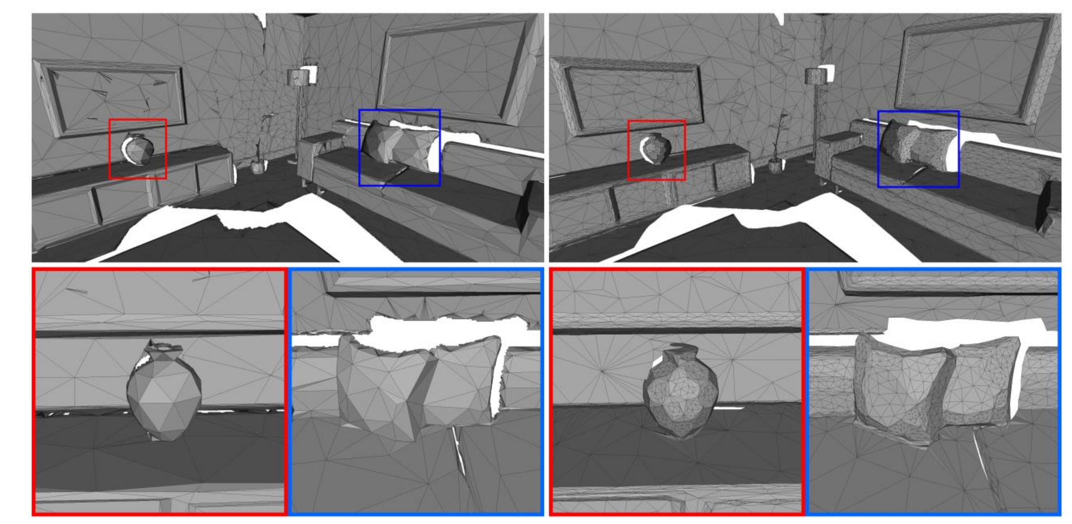
Initial planes



Merged planes

2. Mesh simplification based on planes

Two-step simplification using QEM: firstly inner plane areas separately and entire borders next.



QEM on entire mesh | Ours

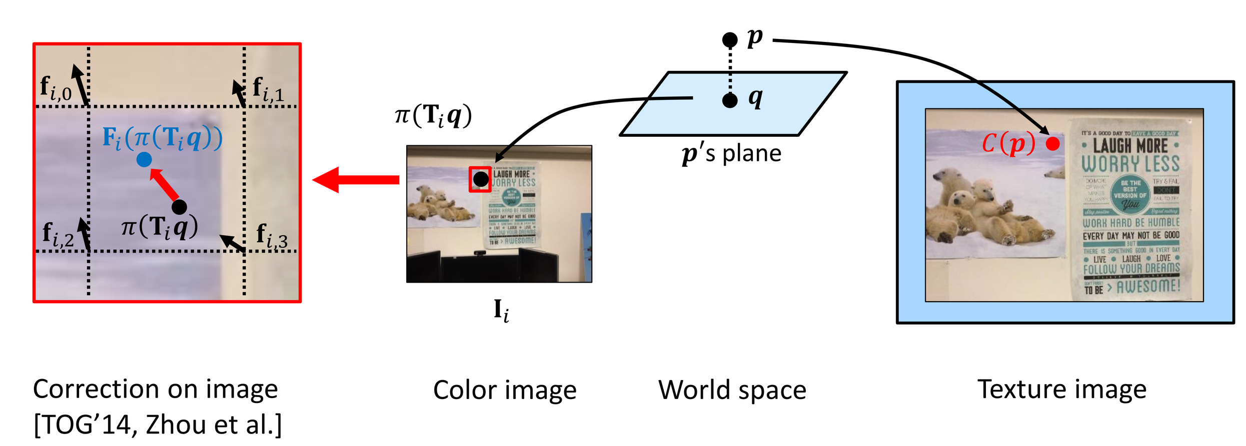
3. Plane, texture and pose optimization

Maximize photometric consistency between texel colors across frames by optimizing camera poses ($\mathbf{T}$), plane parameters ($\mathbf{\Phi}$), textures ($\mathbf{C}$) and image correction offsets ($\mathbf{F}$).

Without optimization | With optimization

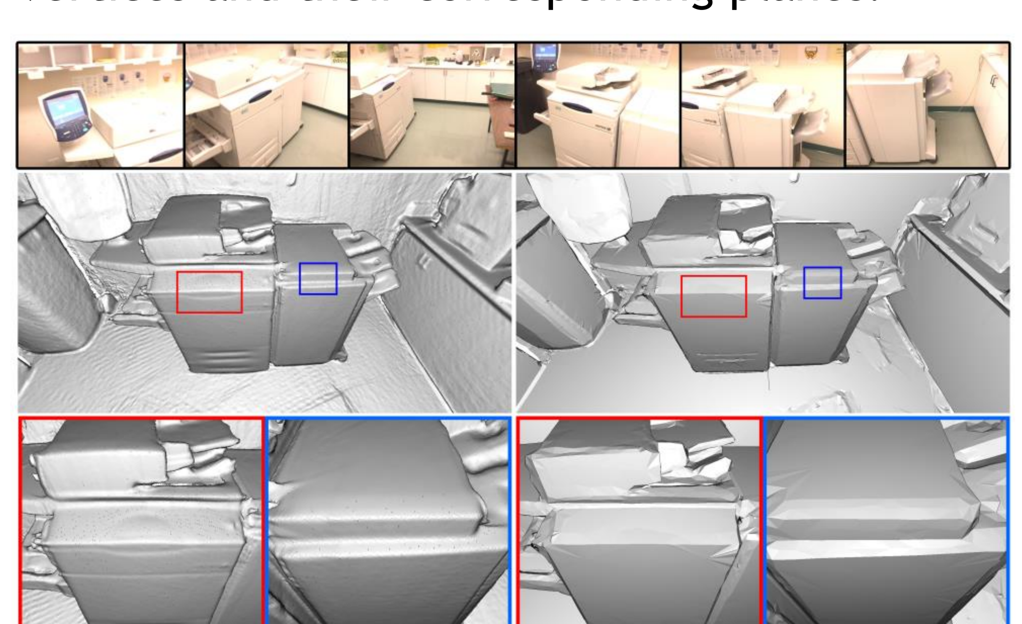
$$E_{tex}(\mathbf{T},\mathbf{\Phi},\mathbf{C},\mathbf{F}) = \underbrace{E_c(\mathbf{T},\mathbf{\Phi},\mathbf{C},\mathbf{F})}_{\text{Photometric consistency}} + \lambda_1 \underbrace{E_p(\mathbf{\Phi})}_{\text{Plane constraint}} + \lambda_2 \underbrace{E_s(\mathbf{F})}_{\text{Image offset constraint}}$$

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



4. Geometry optimization

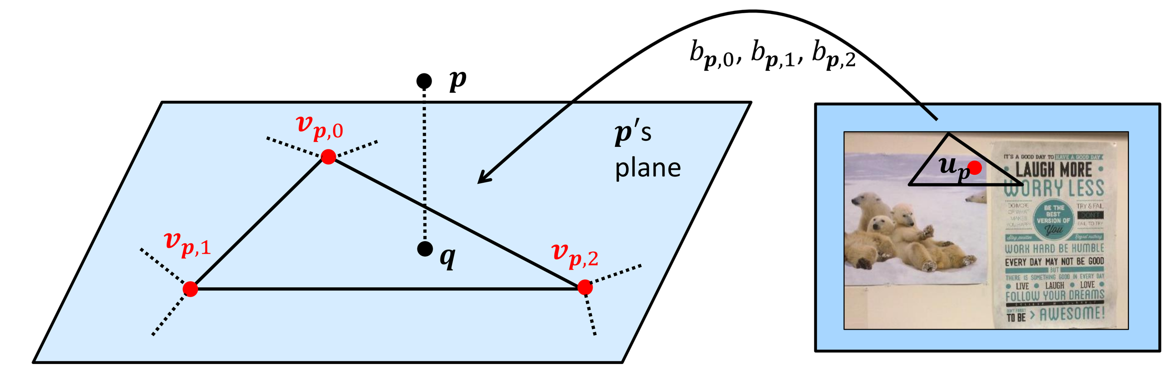
Maximize geometry consistency between mesh vertices and their corresponding planes.



Input fused mesh | Our optimized mesh

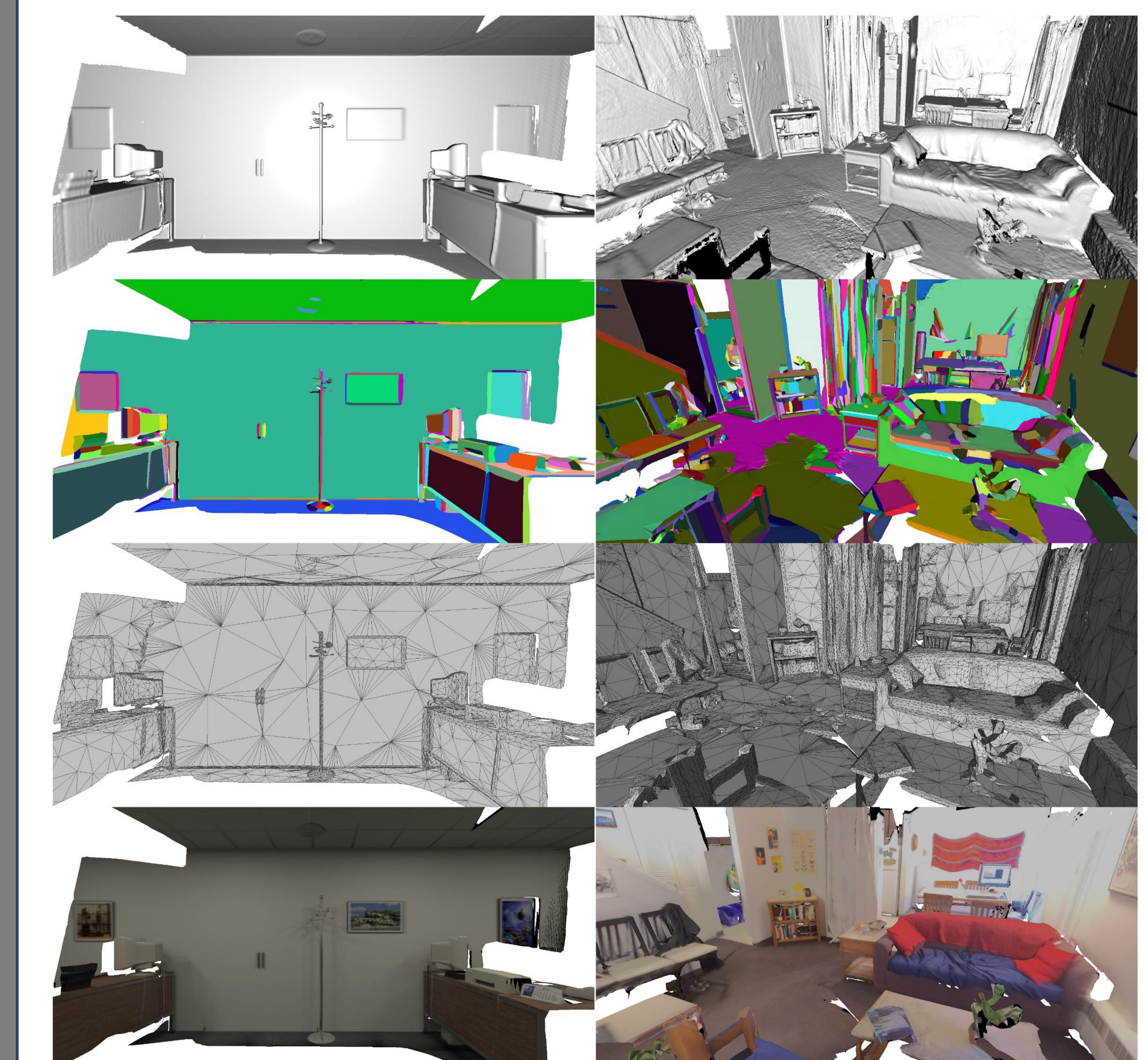
$$E_{vert}(\mathbf{V}) = \underbrace{E_g(\mathbf{V})}_{\text{Geometry-plane consistency}} + \lambda_3 \underbrace{E_t(\mathbf{V})}_{\text{Regularization based on neighbor connectivity}}$$

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

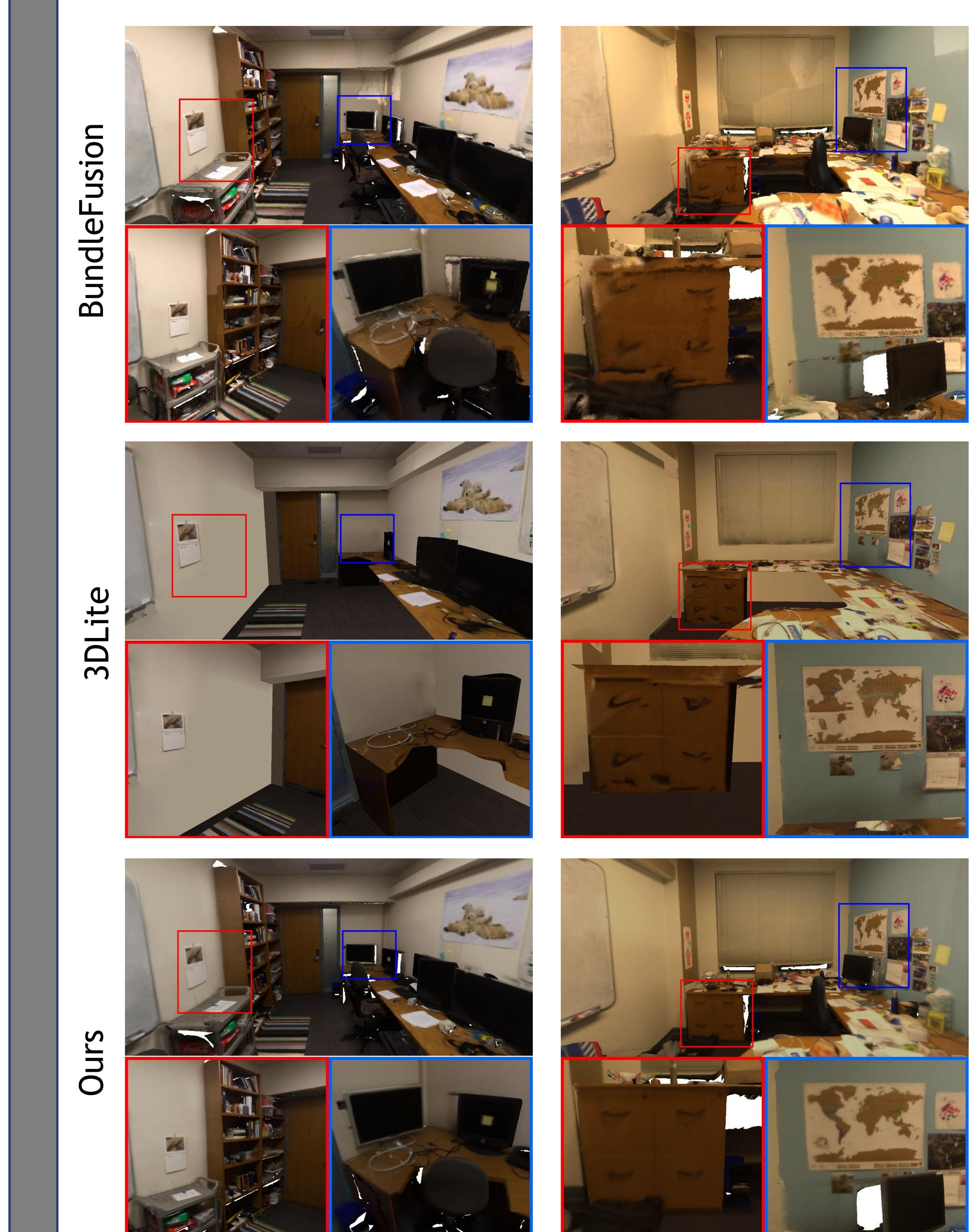


$b_{p,0}, b_{p,1}, b_{p,2}$: $\boldsymbol{u}_p$'s barycentric coordinates inside its triangle on texture image

Results



Results on two sequences: input dense mesh (1st row), planar partition (2nd row), geometry optimization (3rd row) and final textured mesh (4th row).



Comparison between BundleFusion [TOG'17, Dai et al.], 3DLite [TOG'17, Huang et al.] and ours.