Supplementary Material for
Semantic Multi-view Stereo: Jointly Estimating Objects and Voxels

Ali Osman Ulusoy[1] Michael J. Black[1] Andreas Geiger[1,2]
[1]MPI for Intelligent Systems, Tübingen
[2]Computer Vision and Geometry Group, ETH Zürich
{osman.ulusoy,michael.black,andreas.geiger}@tue.mpg.de

Abstract

In this document, we present details of our inference algorithm and additional results. First, we present derivations of the sum-product belief propagation equations. We also present pseudo-code for the inference algorithm. We then present several additional experiments. Our first experiment evaluates the proposed approach with varying model parameters. Second, we present visualizations from our experiment with a small number of images, where our approach significantly improves upon baseline methods. Finally, we present visualizations of the robustness of our approach to approximate input shapes as well as it's ability to combine image and object shape evidence to produce detailed reconstructions.

1. Inference Algorithm Details

Our inference algorithm is based on sum-product particle belief propagation. The main submission presented the probabilistic model but the belief propagation equations and their derivations were omitted due to lack of space. We present detailed derivations in Section 1.1. We then present pseudo-code for our message passing schedule in Section 1.3.

1.1. Message Passing Equations for Sum-product Belief Propagation

We begin by briefly repeating the probabilistic model below for completeness. Please refer to the main submission for the notation and further details.

We formulate volumetric 3D reconstruction as inference in a Markov random field and specify the joint distribution over binary voxel occupancy variables $\mathbf{o}$, continuous voxel color variables $\mathbf{a}$, binary object presence variables $\mathbf{b}$ and continuous object pose variables $\mathbf{p}$ as

$$p(\mathbf{o},\mathbf{a},\mathbf{b},\mathbf{p}) = \frac{1}{Z}\prod_{i\in\mathcal{X}}\varphi_i^o(o_i)\prod_{r\in\mathcal{R}}\psi_r(\mathbf{o}_r,\mathbf{a}_r)\prod_{s\in\mathcal{S}}\left[\varphi_s^b(b_s)\;\varphi_s^p(\mathbf{p}_s)\prod_{q\in\mathcal{Q}_s(\mathbf{p}_s)}\kappa_q(\mathbf{o}_q,b_s,\mathbf{p}_s)\right] \tag{1}$$

where Z denotes the partition function. The potentials φ_i^o, φ_s^b and φ_s^p model the prior beliefs on voxel occupancy, object presence and object pose respectively. The potentials ψ_r and κ_q are high-order ray and raylet potentials respectively.

The general form of the message equation for sum-product belief propagation on factor graphs is given by

$$\mu_{f\to x}(x) = \sum_{\mathcal{X}_f\setminus x}\phi_f(\mathcal{X}_f)\prod_{y\in\mathcal{X}_f\setminus x}\mu_{y\to f}(y) \tag{2}$$

$$\mu_{x\to f}(x) = \prod_{g\in\mathcal{F}_x\setminus f}\mu_{g\to x}(x) \tag{3}$$

where f denotes a factor, x is a random variable, $\mathcal{X}_f$ denotes all variables associated with factor f and $\mathcal{F}_x$ is the set of factors to which variable x is connected.

1.1.1 Message equations for the unary potentials

We first present the factor-to-variable message equations for the unary factors in our MRF, i.e., φ_i^o, φ_s^b, and φ_s^p. These equations are readily given as each factor involves only a single variable:

Voxel Occupancy Prior:

$$\mu_{\varphi_i^o \to o_i}(o_i) = \gamma^{o_i} \, (1-\gamma)^{1-o_i} \tag{4}$$

Object Presence Prior:

$$\mu_{\varphi_s^b \to b_s}(b_s) = \exp(-\lambda_b \, |\mathcal{Q}_s| \, b_s) \tag{5}$$

Object Pose Prior:

$$\mu_{\varphi_s^p \to \mathbf{p}_s}(\mathbf{p}_s) = 1 \tag{6}$$

In this work we do not consider a prior on the object pose. However, this information can be easily integrated into our graphical model.

1.1.2 Message equations for the appearance ray potential

The reader is referred to the supplementary document for [5] for the derivations for the appearance ray potential messages.

1.1.3 Message equations for the raylet potential

The raylet potential message equations are derived similarly to that of the appearance ray potentials. We repeat the raylet potential definition below for completeness.

$$\kappa_q(\mathbf{o}_q, b_s, \mathbf{p}_s) = \begin{cases} \sum_{i=1}^{N_q} o_i^q \prod_{j<i} (1 - o_j^q) \, \eta_i^q(\mathbf{p}_s) & \text{if } b_s = 1 \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

In the following, we derive the message equations for the newly introduced raylet potentials κ_q.

Message to the object model presence variables: We begin with the message from potential to the model presence indicator b_s. Plugging in the potential into Eq. 2 we obtain,

$$\mu_{\kappa_q \to b_s}(b_s) = \int_{\mathbf{p}_s} \sum_{o_1^q} \dots \sum_{o_{N_q}^q} \kappa_q(\mathbf{o}_q, b_s, \mathbf{p}_s) \, \mu(\mathbf{p}_s) \prod_{i=1}^{N_q} \mu(o_i^q) \tag{8}$$

where we have abbreviated the incoming messages as $\mu(\mathbf{p}_s) = \mu_{\mathbf{p}_s \to \kappa_q}$ and $\mu(o_i^q) = \mu_{o_i^q \to \kappa_q}$ for brevity. For $b_s = 0$, we obtain:

$$\mu_{\kappa_q \to b_s}(b_s = 0) = \int_{\mathbf{p}_s} \sum_{o_1^q} \dots \sum_{o_{N_q}^q} 1 \, \mu(\mathbf{p}_s) \prod_{i=1}^{N_q} \mu(o_i^q) = 1 \tag{9}$$

where we assume all incoming messages $\mu(.)$ sum/integrate to 1. We rely on this assumption for the rest of this document to simplify the message equations.

For $b = 1$, we have:

$$\mu_{\kappa_q \to b_s}(b_s = 1) = \sum_{o_1^q} \dots \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \kappa_q(\mathbf{o}_q, b_s, \mathbf{p}_s = 1) \, \mu(\mathbf{p}_s) \prod_{i=1}^{N_q} \mu(o_i^q) \tag{10}$$

Following the derivation of the ray potential message equations, we carry out the summations over the occupancy variables one by one. We start with expanding the summation over o_1.

$$\mu_{\kappa_q \to b_s}(b_s = 1) = \tag{11}$$

$$\mu(o_1^q = 1) \overbrace{\left[\sum_{o_2^q} ... \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \kappa_q(o_1 = 1, o_2, ..., o_{N_q}, b_s, \mathbf{p}_s = 1)\, \mu(\mathbf{p}_s) \prod_{i=2}^{N_q} \mu(o_i^q) \right]}^{(\dagger)} +$$

$$\mu(o_1^q = 0) \underbrace{\left[\sum_{o_2^q} ... \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \kappa_q(o_1 = 0, o_2, ..., o_{N_q}, b_s, \mathbf{p}_s = 1)\, \mu(\mathbf{p}_s) \prod_{i=2}^{N_q} \mu(o_i^q) \right]}_{(\ddagger)}$$

The raylet potential in $(\dagger)$ evaluates to $\eta_1^q(\mathbf{p}_s)$. Hence, we have,

$$(\dagger) = \sum_{o_2^q} ... \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \eta_1^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \prod_{i=2}^{N_q} \mu(o_i^q) \tag{12}$$

$$= \left[\int_{\mathbf{p}_s} \eta_1^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \right] \sum_{o_2^q} ... \sum_{o_{N_q}^q} \prod_{i=2}^{N_q} \mu(o_i^q) = \int_{\mathbf{p}_s} \eta_1^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \tag{13}$$

To evaluate $(\ddagger)$, we expand the summation over o_2 as follows:

$$(\ddagger) = \tag{14}$$

$$\mu(o_2^q = 1) \overbrace{\left[\sum_{o_3^q} ... \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \kappa_q(o_1 = 0, o_2 = 1, o_3, ..., o_{N_q}, b_s, \mathbf{p}_s = 1)\, \mu(\mathbf{p}_s) \prod_{i=3}^{N_q} \mu(o_i^q) \right]}^{(\square)} +$$

$$\mu(o_2^q = 0) \underbrace{\left[\sum_{o_3^q} ... \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \kappa_q(o_1 = 0, o_2 = 0, o_3, ..., o_{N_q}, b_s, \mathbf{p}_s = 1)\, \mu(\mathbf{p}_s) \prod_{i=3}^{N_q} \mu(o_i^q) \right]}_{(\triangle)}$$

We can simplify $(\square)$ similar to the way $(\dagger)$ was simplified. Namely, the raylet potential inside $(\square)$ evaluates to $\eta_2^q(\mathbf{p}_s)$. We have,

$$(\dagger) = \sum_{o_3^q} ... \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \eta_2^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \prod_{i=3}^{N_q} \mu(o_i^q) \tag{15}$$

$$= \left[\int_{\mathbf{p}_s} \eta_2^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \right] \sum_{o_3^q} ... \sum_{o_{N_q}^q} \prod_{i=3}^{N_q} \mu(o_i^q) = \int_{\mathbf{p}_s} \eta_2^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \tag{16}$$

Plugging ($\square$) into ($\ddagger$), we get

$$(\ddagger) = \mu(o_2^q = 1)\left[\int_{\mathbf{p}_s} \eta_2^q(\mathbf{p}_s)\,\mu(\mathbf{p}_s)\right] + \tag{17}$$

$$\mu(o_2^q = 0)\underbrace{\left[\sum_{o_3^q}\dots\sum_{o_{N_q}^q}\int_{\mathbf{p}_s}\kappa_q(o_1=0,o_2=0,o_3,\dots,o_{N_q},b_s,\mathbf{p}_s=1)\,\mu(\mathbf{p}_s)\prod_{i=3}^{N_q}\mu(o_i^q)\right]}_{(\triangle)}$$

Plugging the simplified ($\dagger$) and ($\ddagger$) back into Eq. 11, we get

$$\mu_{\kappa_q\to b_s}(b_s=1) = \mu(o_1^q=1)\left[\int_{\mathbf{p}_s}\eta_1^q(\mathbf{p}_s)\,\mu(\mathbf{p}_s)\right] + \tag{18}$$

$$\mu(o_1^q=0)\mu(o_2^q=1)\left[\int_{\mathbf{p}_s}\eta_2^q(\mathbf{p}_s)\,\mu(\mathbf{p}_s)\right] + \mu(o_1^q=0)\mu(o_2^q=0)\,(\triangle) \tag{19}$$

($\triangle$) can be simplified similarly and so on until all summations over the occupancy variables are expanded. We finally obtain

$$\mu_{\kappa_q\to b_s}(b_s=1) = \sum_{i=1}^{N_q}\mu(o_i^q=1)\prod_{j<i}\mu(o_j^q=0)\left[\int_{\mathbf{p}_s}\eta_i^r(\mathbf{p}_s)\mu(\mathbf{p}_s).\right] \tag{20}$$

Message to the object pose variables: Plugging in the potential into Eq. 2 we obtain,

$$\mu_{\kappa_q\to\mathbf{p}_s}(\mathbf{p}_s) = \sum_{b_s}\sum_{o_1^q}\dots\sum_{o_{N_q}^q}\kappa_q(\mathbf{o}_q,b_s,\mathbf{p}_s)\prod_{i=1}^{N_q}\mu(o_i^q)\,\mu(b_s) \tag{21}$$

Expanding the summation over the binary variable b_s, we obtain

$$\begin{aligned}\mu_{\kappa_q\to\mathbf{p}_s}(\mathbf{p}_s) =& \mu(b_s=0)\left[\sum_{o_1^q}\dots\sum_{o_{N_q}^q}\kappa_q(\mathbf{o}_q,b_s=0,\mathbf{p}_s)\prod_{i=1}^{N_q}\mu(o_i^q)\right] + \\ & \mu(b_s=1)\left[\sum_{o_1^q}\dots\sum_{o_{N_q}^q}\kappa_q(\mathbf{o}_q,b_s=1,\mathbf{p}_s)\prod_{i=1}^{N_q}\mu(o_i^q)\right] \qquad (22)\\ =& \mu(b_s=0)\left[\sum_{o_1^q}\dots\sum_{o_{N_q}^q}1\prod_{i=1}^{N_q}\mu(o_i^q)\right] + \\ & \mu(b_s=1)\left[\sum_{o_1^q}\dots\sum_{o_{N_q}^q}\kappa_q(\mathbf{o}_q,b_s=1,\mathbf{p}_s)\prod_{i=1}^{N_q}\mu(o_i^q)\right] \qquad (23)\\ =& \mu(b_s=0)+\mu(b_s=1)\underbrace{\left[\sum_{o_1^q}\dots\sum_{o_{N_q}^q}\kappa_q(\mathbf{o}_q,b_s=1,\mathbf{p}_s)\prod_{i=1}^{N_q}\mu(o_i^q)\right]}_{(\triangledown)} \qquad (24)\end{aligned}$$

($\triangledown$) can be simplified similarly as in the derivation of the object presence variable b, by expanding the summations over the occupancy variables one by one. We obtain,

$$\mu_{\kappa_q\to\mathbf{p}_s}(\mathbf{p}_s) = \mu(b_s=0)+\mu(b_s=1)\left[\sum_{i=1}^{N_q}\mu(o_i^q=1)\prod_{j<i}\mu(o_j^q=0)\,\eta_i^q(\mathbf{p}_s)\right] \tag{25}$$

Message to the voxel occupancy variables: Plugging in the potential into Eq. 2 we obtain,

$$\mu_{\kappa_q \to o_i^q}(o_i^q) = \sum_{b_s} \sum_{o_1^q} ... \sum_{o_{i-1}^q} \sum_{o_{i+1}^q} ... \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \kappa_q(\mathbf{o}_q, b_s, \mathbf{p}_s) \prod_{\substack{j=1 \\ j \neq i}}^{N_q} \mu(o_j^q)\, \mu(b_s) \tag{26}$$

We begin by expanding the summation over the binary object model presence variable.

$$\begin{aligned}
&\mu_{\kappa_q \to o_i^q}(o_i^q) = \\
&\mu(b_s = 0) \left[\sum_{o_1^q} ... \sum_{o_{i-1}^q} \sum_{o_{i+1}^q} ... \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \kappa_q(\mathbf{o}_q, b_s = 0, \mathbf{p}_s) \prod_{\substack{j=1 \\ j \neq i}}^{N_q} \mu(o_j^q) \right] + \\
&\mu(b_s = 1) \left[\sum_{o_1^q} ... \sum_{o_{i-1}^q} \sum_{o_{i+1}^q} ... \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \kappa_q(\mathbf{o}_q, b_s = 1, \mathbf{p}_s) \prod_{\substack{j=1 \\ j \neq i}}^{N_q} \mu(o_j^q) \right] \\
&= \mu(b_s = 0) + \mu(b_s = 1) \underbrace{\left[\sum_{o_1^q} ... \sum_{o_{i-1}^q} \sum_{o_{i+1}^q} ... \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \kappa_q(\mathbf{o}_q, b_s = 1, \mathbf{p}_s) \prod_{\substack{j=1 \\ j \neq i}}^{N_q} \mu(o_j^q) \right]}_{(\diamond)}
\end{aligned} \tag{27}$$

We simplify $(\diamond)$ similar to the derivations above. For $o_i^q = 1$ we have,

$$\begin{aligned}
(\diamond) &= \sum_{o_1^q} ... \sum_{o_{i-1}^q} \sum_{o_{i+1}^q} ... \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \kappa_q(o_1^q, ..., o_{i-1}^q, o_i^q = 1, o_{i+1}^q, ..., o_{N_q}^q, b_s = 1, \mathbf{p}_s) \prod_{\substack{j=1 \\ j \neq i}}^{N_q} \mu(o_j^q) \\
&= \sum_{j<i} \mu(o_j^q = 1) \prod_{k<j} \mu(o_k^q = 0) \big[\int_{\mathbf{p}_s} \eta_j^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \big] \\
&+ \prod_{k<i} \mu(o_k^q = 0) \left[\int_{\mathbf{p}_s} \eta_i^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \right]
\end{aligned} \tag{28}$$

For $o_i^q = 0$ we have,

$$\begin{aligned}
(\diamond) &= \sum_{o_1^q} ... \sum_{o_{i-1}^q} \sum_{o_{i+1}^q} ... \sum_{o_{N_q}^q} \int_{\mathbf{p}_s} \kappa_q(o_1^q, ..., o_{i-1}^q, o_i^q = 0, o_{i+1}^q, ..., o_{N_q}^q, b_s = 1, \mathbf{p}_s) \prod_{\substack{j=1 \\ j \neq i}}^{N_q} \mu(o_j^q) \\
&= \sum_{j<i} \mu(o_j^q = 1) \prod_{k<j} \mu(o_k^q = 0) \big[\int_{\mathbf{p}_s} \eta_j^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \big] + \sum_{j>i} \mu(o_j^q = 1) \prod_{\substack{k<j \\ k \neq i}} \mu(o_k^q = 0) \big[\int_{\mathbf{p}_s} \eta_j^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \big]
\end{aligned} \tag{29}$$

In summary, the message to the occupancy variables can be written as,

$$\begin{aligned}
\mu_{\kappa_q \to o_i^q}(o_i^q = 1) =& \mu(b_s = 0) + \mu(b_s = 1) \\
&\cdot \left[\sum_{j<i} \mu(o_j^q = 1) \prod_{k<j} \mu(o_k^q = 0) \big[\int_{\mathbf{p}_s} \eta_j^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \big] + \prod_{k<i} \mu(o_k^q = 0) \big[\int_{\mathbf{p}_s} \eta_i^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \big] \right]
\end{aligned} \tag{30}$$

$$\begin{aligned}
\mu_{\kappa_q \to o_i^q}(o_i^q = 0) =& \mu(b_s = 0) + \mu(b_s = 1) \\
&\cdot \left[\sum_{j<i} \mu(o_j^q = 1) \prod_{k<j} \mu(o_k^q = 0) \big[\int_{\mathbf{p}_s} \eta_j^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \big] + \sum_{j>i} \mu(o_j^q = 1) \prod_{\substack{k<j \\ k \neq i}} \mu(o_k^q = 0) \big[\int_{\mathbf{p}_s} \eta_j^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \big] \right]
\end{aligned} \tag{31}$$

1.2. Particle Belief Propagation

The integral equations that arise in the message passing equations (Eq. 20+30+31) do not admit closed form solutions. As discussed in the main submission, we follow a particle based strategy [3] and maintain a sample distribution to approximate the continuous state space of $\mathbf{p}$. This discretization allows Monte Carlo estimates of the integral equations.

Let $\{\mathbf{p}_s^{(1)}, \ldots, \mathbf{p}_s^{(K)}\}$ denote the set of particles, and let $\omega(\mathbf{p}_s)$ denote the distribution obtained using a kernel density estimator on $\{\mathbf{p}_s^{(1)}, \ldots, \mathbf{p}_s^{(K)}\}$. We use this discretization to approximate the following message equations.

Message to the object model presence variables:

$$\begin{aligned}\mu_{\kappa_q \to b_s}(b_s) &= \int_{\mathbf{p}_s} \sum_{o_1^q} \ldots \sum_{o_{N_q}^q} \kappa_q(\mathbf{o}_q, b_s, \mathbf{p}_s)\, \mu(\mathbf{p}_s) \prod_{i=1}^{N_q} \mu(o_i^q) \\ &\approx \frac{1}{K} \sum_{k=1}^{K} \sum_{o_1^q} \ldots \sum_{o_{N_q}^q} \kappa_q(\mathbf{o}_q, b_s, \mathbf{p}_s) \frac{\mu(\mathbf{p}_s^{(k)})}{\omega(\mathbf{p}_s^{(k)})} \prod_{i=1}^{N_q} \mu(o_i^q) \end{aligned} \tag{32}$$

Now we evaluate the specific cases $b_s = 0$ and $b_s = 1$:

$$\mu_{\kappa_q \to b_s}(b_s = 0) \approx \frac{1}{K} \sum_{k=1}^{K} \sum_{o_1^q} \ldots \sum_{o_{N_q}^q} \kappa_q(\mathbf{o}_q, b_s, \mathbf{p}_s = 0) \frac{\mu(\mathbf{p}_s^{(k)})}{\omega(\mathbf{p}_s^{(k)})} \prod_{i=1}^{N_q} \mu(o_i^q) = \frac{1}{K} \sum_{k=1}^{K} \frac{\mu(\mathbf{p}_s^{(k)})}{\omega(\mathbf{p}_s^{(k)})} \tag{33}$$

$$\begin{aligned}\mu_{\kappa_q \to b_s}(b_s = 1) &= \sum_{i=1}^{N_q} \mu(o_i^q = 1) \prod_{j<i} \mu(o_j^q = 0) \left[\int_{\mathbf{p}_s} \eta_i^r(\mathbf{p}_s) \mu(\mathbf{p}_s). \right] \\ &\approx \sum_{i=1}^{N_q} \mu(o_i^q = 1) \prod_{j<i} \mu(o_j^q = 0) \left[\frac{1}{K} \sum_{k=1}^{K} \frac{\mu(\mathbf{p}_s^{(k)})}{\omega(\mathbf{p}_s^{(k)})} \eta_i^r(\mathbf{p}_s^{(k)}) \right] \end{aligned} \tag{34}$$

Message to the voxel occupancy variables: Similar to the approximation above, the messages to the voxel occupancies Eq. 30+31 contain the integral $\int_{\mathbf{p}_s} \eta_i^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s)$ which is approximated as follows:

$$\int_{\mathbf{p}_s} \eta_i^q(\mathbf{p}_s)\, \mu(\mathbf{p}_s) \approx \frac{1}{K} \sum_{k=1}^{K} \frac{\mu(\mathbf{p}_s^{(k)})}{\omega(\mathbf{p}_s^{(k)})} \eta_i^r(\mathbf{p}_s^{(k)}) \tag{35}$$

1.3. Inference Algorithm Pseudo-code

In the following, we present pseudo-code our inference algorithm. The general algorithm is presented in Algorithm 1.

In an offline stage, we compute the truncated signed distance function (TSDF) of each input object shape model. We assume the shapes are encoded as meshes. We position the objects in the center of the scene for this computation. This pre-computation allows evaluating the TSDF of an object in any pose simply by a look-up, which is very fast. Since our algorithm evaluates the TSDF of each object for thousands of poses during inference, this pre-computation enables tractable inference.

For the coarse pose sampling (line 2) in the proposal generation algorithm (Algorithm 3), we make use of the ground plane knowledge in our datasets. In particular, we estimate the ground plane by robustly fitting a plane to the point cloud generated by structure-from-motion. For purposes of the proposal generation, the pose of each object is modeled as the translation on the XY plane, rotation around the Z axis (perpendicular to the ground plane) as well as scale. We further assume each shape model is roughly at the correct scale. For the aerial datasets, we used the geolocation information in the Trimble Warehouse. For the LIVINGROOM dataset, we manually scaled the models to approximately the right scale.

Algorithm 1 Inference algorithm

```
1: procedure INFERENCE
2:     Shuffle images.
3:     Perform an initial round of message passing for the appearance ray potentials as proposed in [5].
4:     while not converged do
5:         GENERATE POINT CLOUD()
6:         for each input shape model do
7:             GENERATE PARTICLES()
8:             Perform message passing for all raylet potentials.
           REFINE OCTREE()
9:         Perform a round of message passing for the appearance ray potentials [5].
```

Algorithm 2 Generate Point Cloud Procedure

```
1: procedure GENERATE POINT CLOUD
2:     for each octree leaf do
3:         if probability (belief) of voxel occupancy > 0.3 then
4:             Add voxel center to point cloud.
                                                          ▷ Sparsification
5:     for each point in the point cloud do
6:         neighbors ← all elements in the point cloud within ϵ distance.
7:         Remove neighbors from the point cloud.
```

Algorithm 3 Generate Object Pose Proposals (Particles) Procedure

```
1: procedure GENERATE PARTICLES
2:     Coarsely sample the pose space.
3:     Evaluate Eq. 9 (in the original doc.) for each proposal and keep the best modes.
4:     samples ← ∅
5:     for each mode do
6:         samples ← ⋃ Metropolis-Hastings (MH) sampling with a Gaussian proposal centered at the mode. We run MH
   for 10K iterations.
7:     Sort samples according to their energy (Eq. 9 in the original doc.)
8:     Remove similar samples and retain at most K = 64 particles.
```

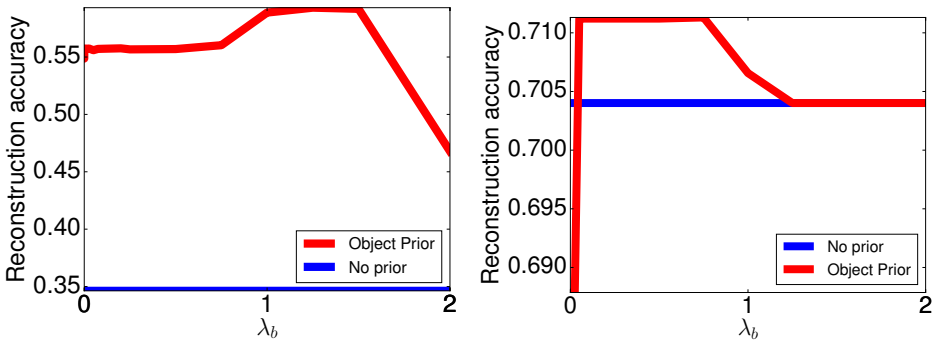
Algorithm 4 Octree Refinement Procedure

procedure REFINE OCTREE
  for each octree leaf **do**
    if probability (belief) of voxel occupancy > 0.3 **then** ▷ empirically chosen threshold
      Subdivide octree leaf into eight children.
      Initialize the voxel occupancy and appearance messages to uniform.

2. Parameter Experiments

Our probabilistic model contains two main parameters: λ_b and λ_p. $\lambda_b > 0$ expresses our belief about the presence of an input object shape, e.g. a high λ_b means a low prior belief on the presence of the object. Although this parameter could be tuned per object shape (according to semantic information available), we use $\lambda_b = 0.75$ for all objects in our experiments. Empirically we found this setting to work well. The other parameter $\lambda_p > 0$ controls the strength of the likelihood for the raylet potentials. A small λ_p yields a smoother likelihood, whereas a high λ_p results in a peaked likelihood. For all experiments, we use $\lambda_p = 8$.

In this section, we present quantitative evaluations with varying model parameters λ_b and λ_p. We use the DOWNTOWN dataset for this evaluation and the evaluation protocol explained in the main submission document. In particular, we test the approach using varying λ_b and λ_p values as well as compare against Ulusoy et al. [5] whose formulation is equivalent to removing the object shape prior from our model and which we refer to as "No prior" in the following. Further, we evaluate both approaches on a small number (9) and a large number (180) of images.



(a) 9 images from DOWNTOWN.

(b) 180 images from DOWNTOWN.

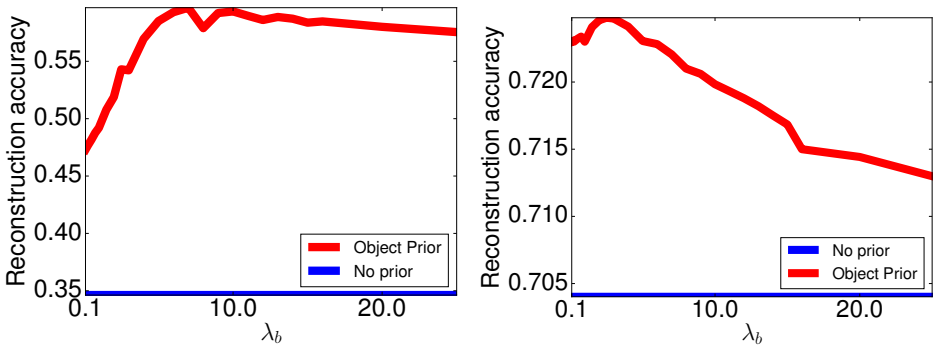
Figure 1: Quantitative evaluation of varying model parameter λ_b and comparison to the baseline approach with no prior [5]. The evaluation was done using the DOWNTOWN dataset. Higher is better.

For varying λ_p, we present the quantitative results in Fig. 1. For small number of images (see Fig. 1a), even a small λ_b helps improve the reconstruction. We found that for such low λ_b, the algorithm chooses to enable almost all input shape models (see Fig 1b in the original submission). Although four of these models are actually not present in the scene, their presence does not decrease performance substantially. As λ_b is increased, these false positive detections are removed and the model enables only the correct 7 models that are present in the scene. This leads to an improvement in performance for $\lambda > 0.75$. Finally, when λ_p is increased even further, i.e. $\lambda_b > 1.5$ the model begins to disable even the correct models, i.e. removing the shape prior. As expected, the performance decreases until all models are turned off in which case the performance is equivalent to the approach with no shape prior [5]. For high number of images, the overall trend is the same as seen in Fig. 1b.

For varying λ_b, we present the quantitative results in Fig. 2 and provide visualizations in Fig. 3+4. It can be observed that for a small number of images (Fig. 3+2a) a small λ_p leads to artifacts around the object shape, due to the wide likelihood function. As λ_p gets larger, i.e. the likelihood gets narrower, the artifacts disappear and the results improve. Note that for $\lambda_p > 10$ the performance starts to decline slowly since the likelihood function becomes very narrow, which does not allow precise pose fitting.

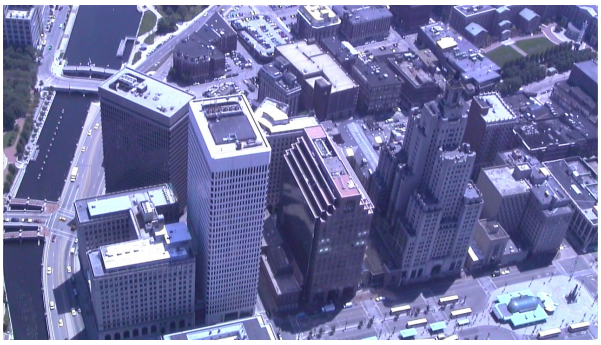
A similar trend can be observed for the large number of images (Fig. 4+2b) as well. For this case though, a small λ_p does not lead to large artifacts around the object shape, since there's significant image evidence which overrides these artifacts.

Moreover, the change in accuracy with respect to varying λ_p is also much smaller compared to the 9 image case.

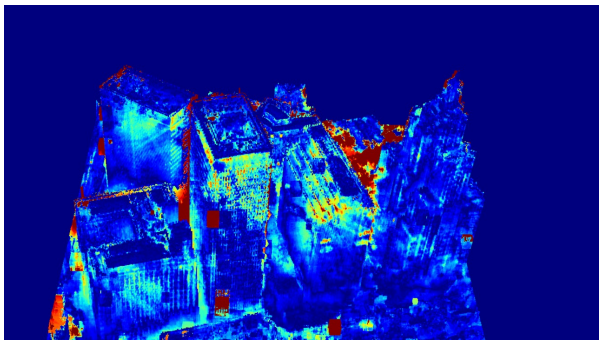


(a) 9 images from DOWNTOWN. (b) 180 images from DOWNTOWN.

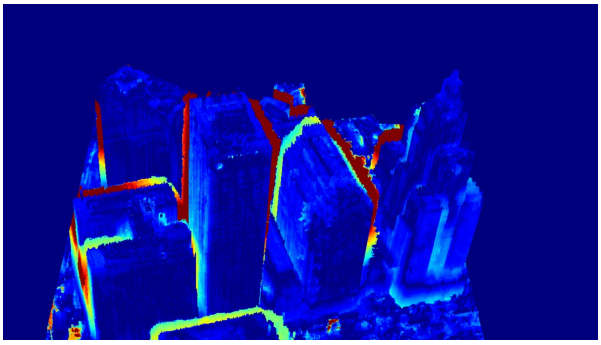
Figure 2: Quantitative evaluation of varying model parameter λ_p and comparison to the baseline approach with no prior [5]. The evaluation was done using the DOWNTOWN dataset. Higher is better.



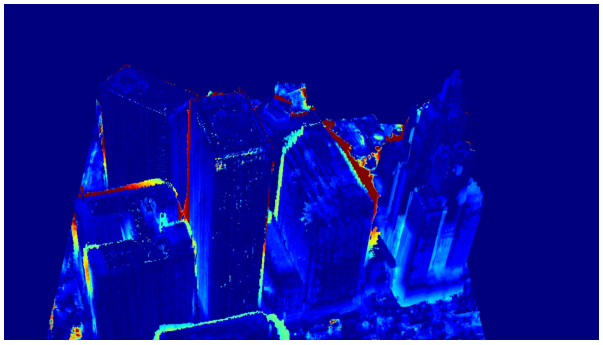
(a) Reference image.



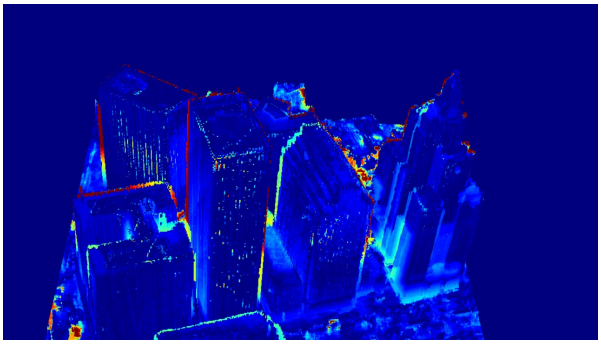
(b) No prior [5].



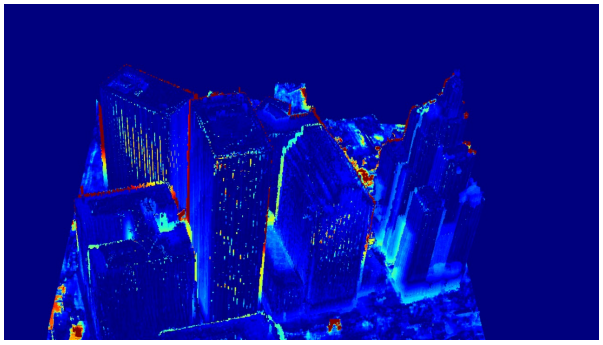
(c) Our shape prior with $\lambda_p = 1$.



(d) Our shape prior with $\lambda_p = 5$.



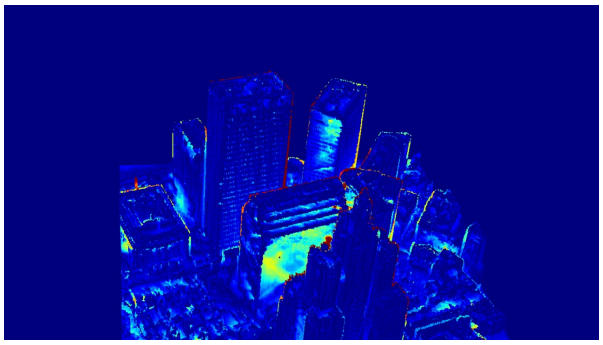
(e) Our shape prior with $\lambda_p = 10$.



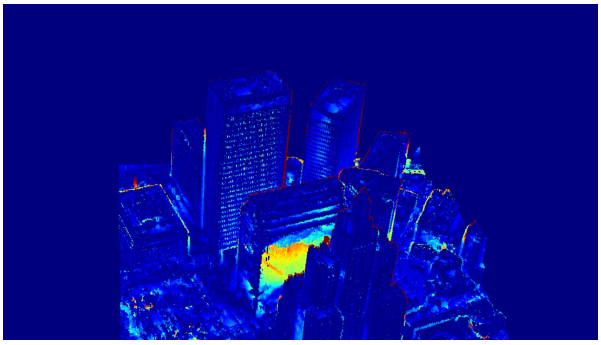
(f) Our shape prior with $\lambda_p = 20$.

Figure 3: (c-f) Visualizations of depth error for varying model parameter λ_p using 9 images from the DOWNTOWN dataset. Cooler colors correspond to lower error.

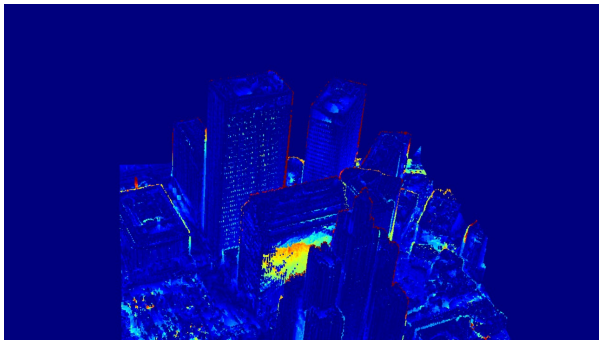
(a) Reference image.



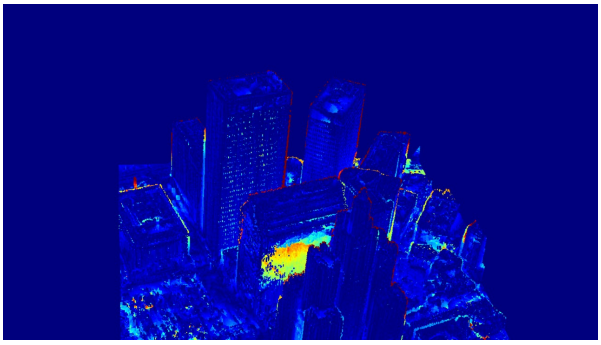
(b) No prior [5].



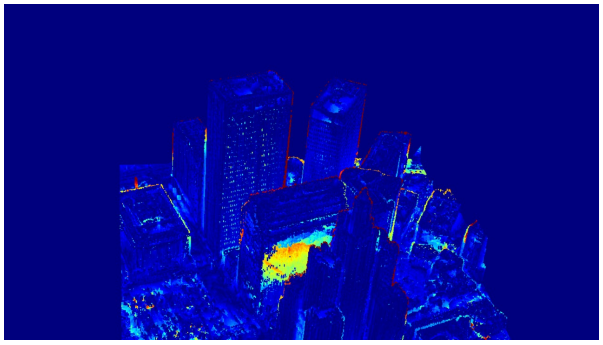
(c) Our shape prior with $\lambda_p = 1$.



(d) Our shape prior with $\lambda_p = 5$.



(e) Our shape prior with $\lambda_p = 10$.



(f) Our shape prior with $\lambda_p = 20$.

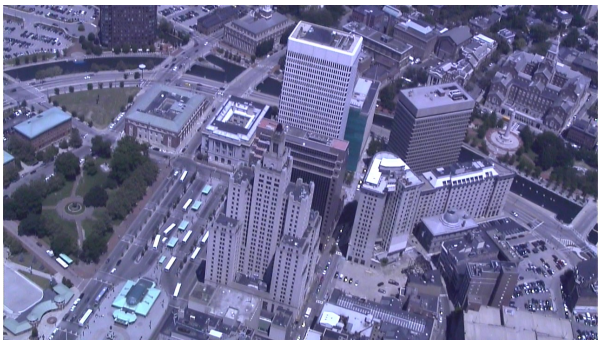
Figure 4: (c-f) Visualizations of depth error for varying model parameter λ_p using 180 images from the DOWNTOWN dataset. Cooler colors correspond to lower error.

3. Small number of images – Additional examples

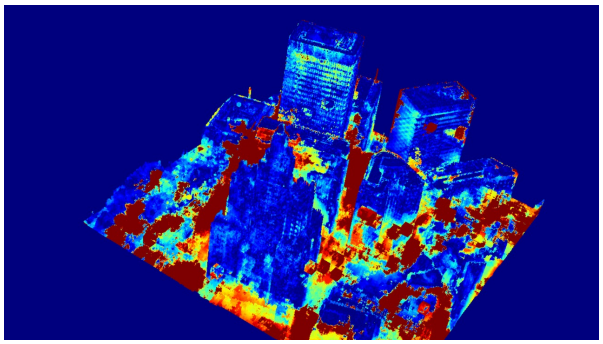
For a small number ($\sim$10) of images, the object shape prior achieves significant improvements over the baseline without any prior. Fig 5 in the original submission provides an example. We show further visualizations of the depth map errors in Fig. 5+6+7.

The results are consistent with Fig 5 of the original text. Our approach (Fig. 5d+6d+7d) improves accuracy near building surfaces but notably also in other parts of the scene with respect to the baseline (Fig. 5b+6b+7b). In particular, our method exploits the geometric knowledge induced by the prior to refine free-space areas and visibility constraints, leading to higher accuracy also in regions for which no shape priors are available.

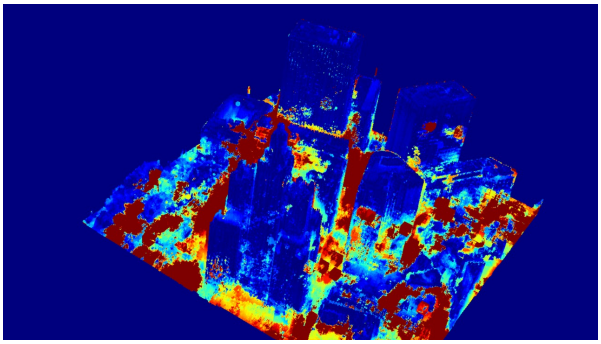
We further demonstrate the power of the proposed joint inference scheme by comparing to a baseline which bootstraps the reconstruction using [5] and fuses this information with the 3D shape models using one iteration of raylet-to-voxel message passing. Note that this is similar to the existing approach of fitting a 3D shape model to the reconstruction [1, 2, 4, 6]. As shown in Fig. 5c+6c+7c, the result is significantly worse compared to our full model Fig. 5d+6d+7d which integrates image evidence, shape priors and visibility constraints in a principled probabilistic fashion.



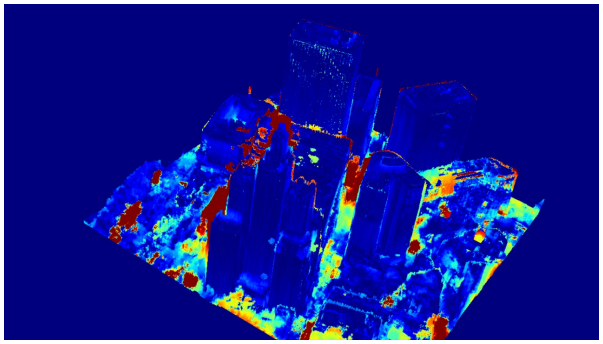
(a) Reference image.



(b) No prior [5]



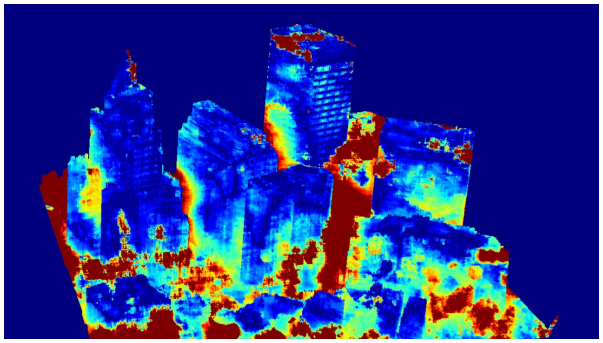
(c) Shape prior without joint inference.



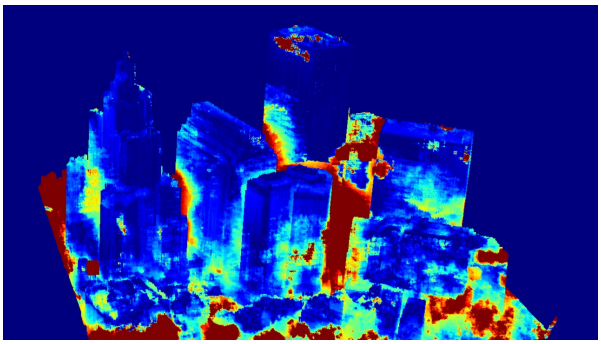
(d) Proposed shape prior.

Figure 5: Visualizations of depth error for the DOWNTOWN2 dataset. Cooler colors correspond to lower error.

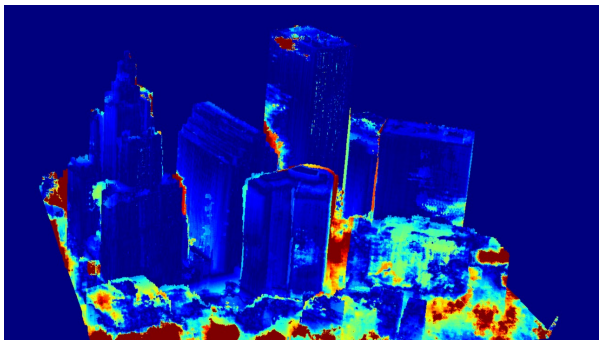
(a) Reference image.



(b) No prior [5]

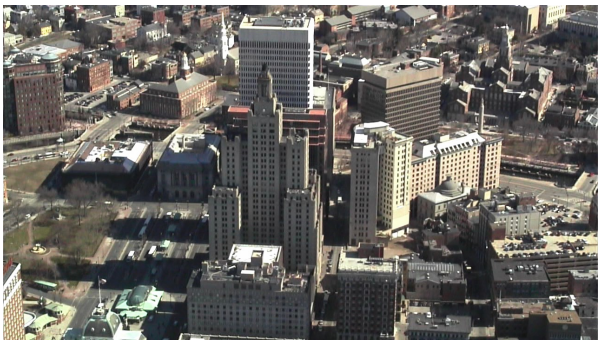


(c) Shape prior without joint inference.

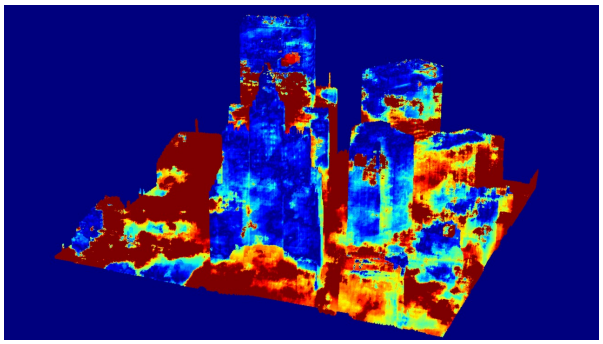


(d) Proposed shape prior.

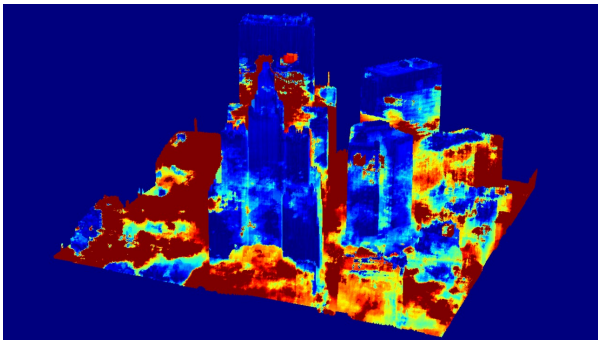
Figure 6: Visualizations of depth error for the DOWNTOWN2 dataset. Cooler colors correspond to lower error.



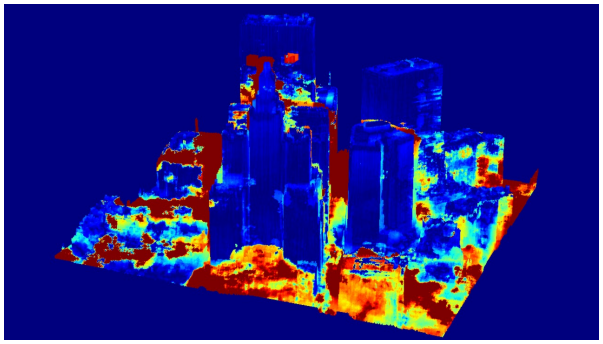
(a) Reference image.



(b) No prior [5]

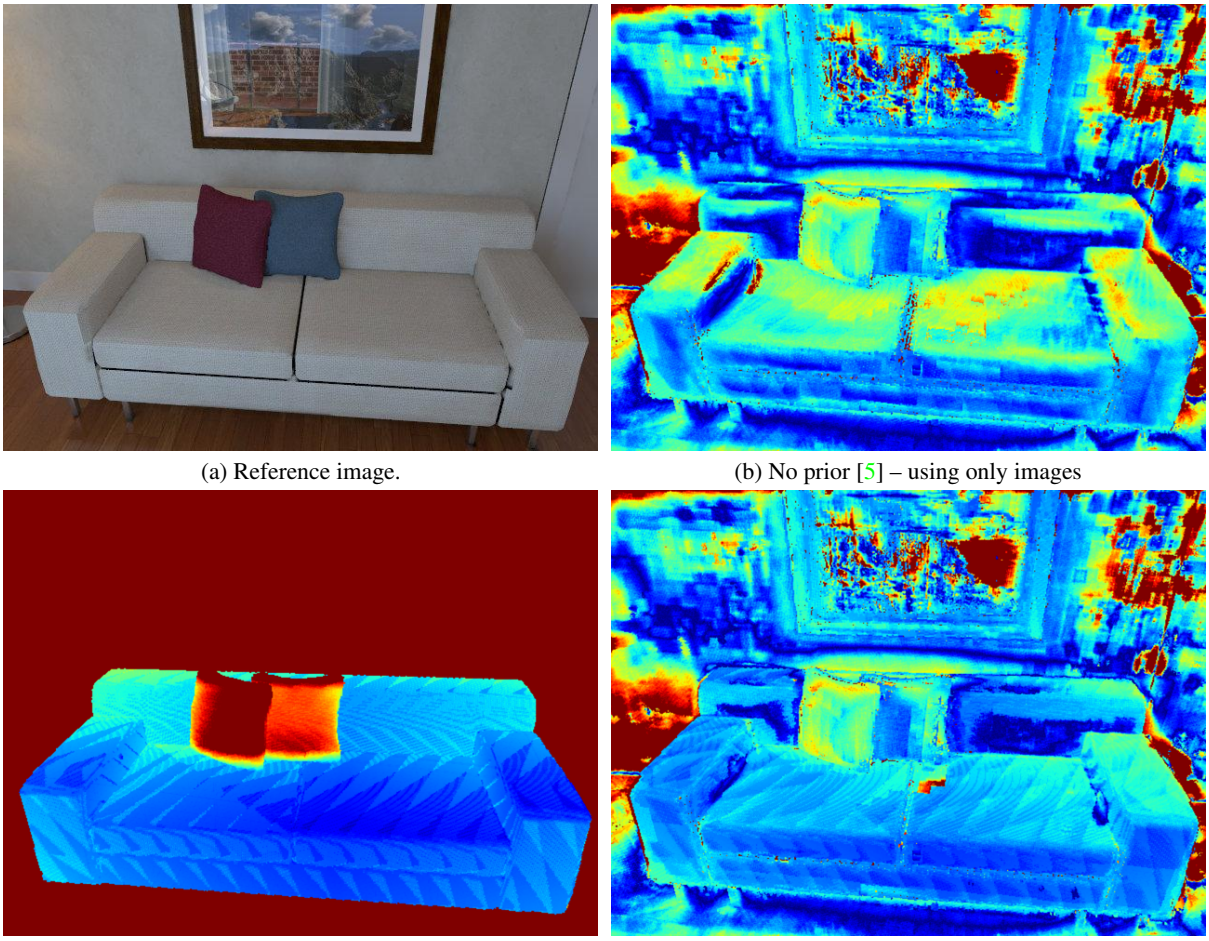


(c) Shape prior without joint inference.



(d) Proposed shape prior.

Figure 7: Visualizations of depth error for the DOWNTOWN2 dataset. Cooler colors correspond to lower error.



(a) Reference image.

(b) No prior [5] – using only images

(c) Using only the shape model.

(d) Proposed shape prior – combining image and input shape.

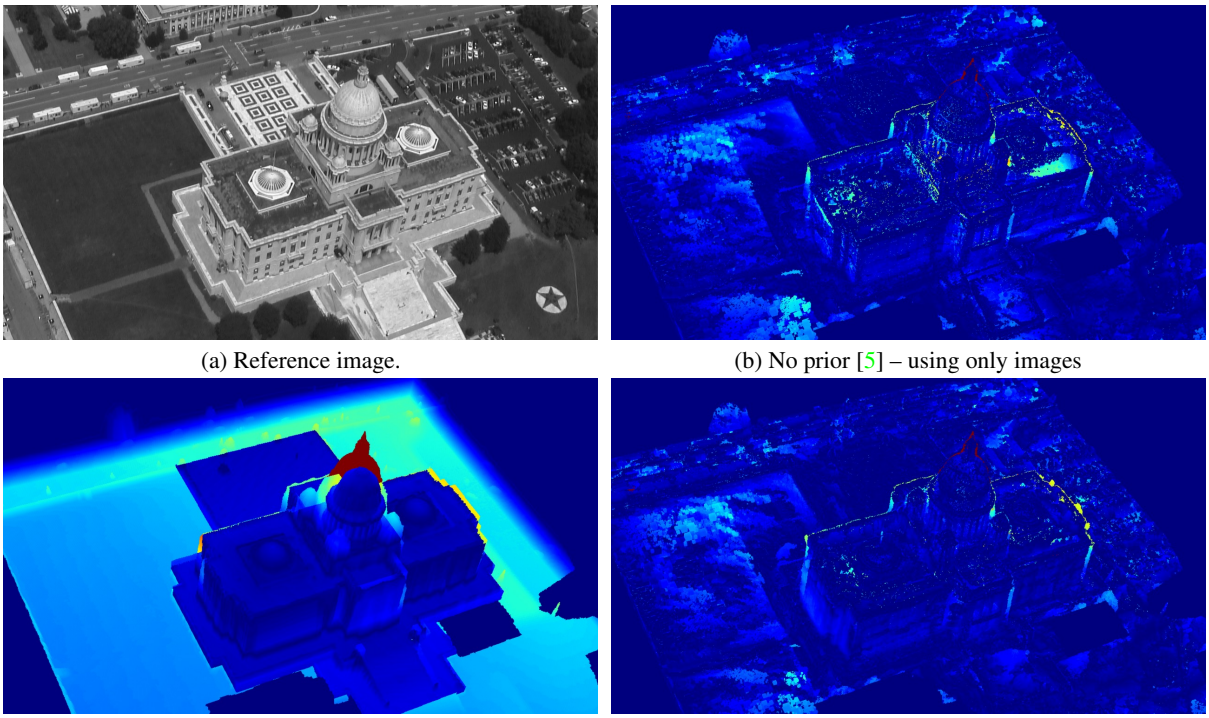
Figure 8: Visualizations of depth error for the LIVINGROOM dataset. Cooler colors correspond to lower error.

4. Combining image and shape evidence – Additional examples

Our method is able to combine image evidence and the input shape models to produce detailed reconstructions. This section presents visualizations that show the benefit of combining image evidence and input shape models.

The scene in Fig. 8 contains a sofa and two pillows on the sofa. Fig. 8b presents the depth errors using images only, i.e. the method of Ulusoy et al [5]. Input object shape is the sofa as presented in Fig. 6a of the original submission. Fitting this shape to the 3D reconstruction allows evaluating the reconstruction only using the sofa shape. The depth error is shown in Fig. 8c. As expected, there are large errors on the pillows which are not part of the sofa shape. Our approach (Fig. 8d) combines the input shape with image evidence to reconstruct both the sofa and the pillow, improving over both Fig. 8b and Fig. 8c.

An example from the CAPITOL is presented in Fig. 9. Fig. 9b visualizes the errors of the approach using only images [5]. Note that the approach yields errors on the reflective rooftop. The input shape model (see fig 7a in the original submission) is missing the small tower next to the copula and furthermore, the copula shape is incorrect as can be seen in Fig. 9c. Our approach (see Fig. 9d) combines the images with the input shape model to improve over both Fig. 9c and Fig. 9b.



(a) Reference image.

(b) No prior [5] – using only images

(c) Using only the shape model.

(d) Proposed shape prior – combining image and input shape.

Figure 9: Visualizations of depth error for the CAPITOL dataset. Cooler colors correspond to lower error.

References

[1] S. Bao, M. Chandraker, Y. Lin, and S. Savarese. Dense object reconstruction with semantic priors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 11

[2] A. Dame, V. Prisacariu, C. Ren, and I. Reid. Dense reconstruction using 3D object shape priors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 11

[3] A. Ihler and D. McAllester. Particle belief propagation. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009. 6

[4] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. SLAM++: simultaneous localisation and mapping at the level of objects. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 11

[5] A. O. Ulusoy, A. Geiger, and M. J. Black. Towards probabilistic volumetric reconstruction using ray potentials. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2015. 2, 7, 8, 9, 10, 11, 12, 13, 14, 15

[6] C. Zhou, F. Güney, Y. Wang, and A. Geiger. Exploiting object similarity in 3d reconstruction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2015. 11