Supplementary Document for Patches, Planes and Probabilities: A Non-local Prior for Volumetric 3D Reconstruction

Ali Osman Ulusoy Michael J. Black Andreas Geiger
Max Planck Institute for Intelligent Systems, Tübingen, Germany
{osman.ulusoy,michael.black,andreas.geiger}@tue.mpg.de

Abstract

This supplementary document presents derivations for the proposed sum-product belief propagation algorithm, pseudocode of the inference algorithm, the derivations of our depth-map prediction method as well as additional experiments. First, we present the message derivations of the sum-product belief propagation algorithm which were omitted in the original document. We then present the pseudocode of our inference algorithm, and in particular the message passing scheme. Besides, we show how Bayes optimal depth predictions can be obtained under our probabilistic model. Finally, we present a number of additional experiments. In particular, we present results by varying the parameters for the model with pairwise smoothness potentials. Next, we present an evaluation for the BARUS&HOLLEY *dataset which excludes the tree regions where the LIDAR ground truth is not accurate and show that our algorithm outperforms previous algorithms. Finally, we present an experiment using a uniform prior over plane orientations as opposed to the Manhattan world prior that we utilize in the paper.*

1. Message Equations for Sum-product Belief Propagation

This section presents the message equations and their derivations that were omitted in the main submission due to lack of space. We refer the reader to the submission file for the notation and the probabilistic model.

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

$$\mu_{f\to x}(x) = \sum_{\mathcal{X}_f \setminus x} \phi_f(\mathcal{X}_f) \prod_{y \in \mathcal{X}_f \setminus x} \mu_{y\to f}(y) \tag{1}$$

$$\mu_{x\to f}(x) = \prod_{g \in \mathcal{F}_x \setminus f} \mu_{g\to x}(x) \tag{2}$$

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. Below, we repeat the factor graph of our model for completeness.

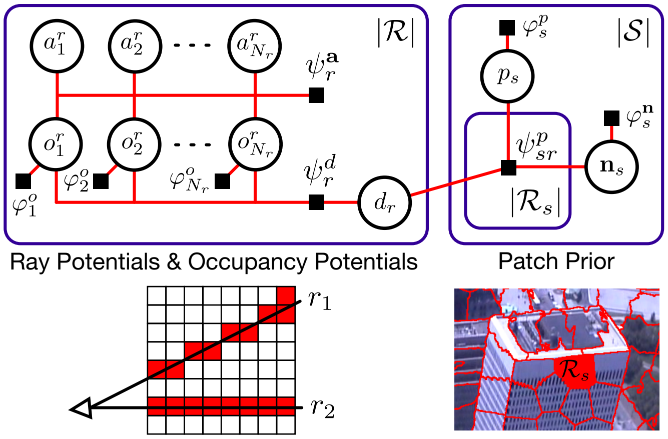


Figure 1: Factor graph of our probabilistic model.

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^p, and $\varphi_s^{\mathbf{n}}$. These equations are readily given by Eq. 1 as each factor involves only a single variable:

Voxel Occupancy Prior:

$$\mu_{\varphi_i^o \to o_i}(o_i) = \gamma^{o_i}\,(1-\gamma)^{1-o_i} \tag{3}$$

Planarity Potential:

$$\mu_{\varphi_s^p \to p_s}(p_s) = \exp(\lambda_s\,|\mathcal{R}_s|\,p_s) \tag{4}$$

Normal Potential:

$$\mu_{\varphi_s^{\mathbf{n}} \to \mathbf{n}_s}(\mathbf{n}_s) = \sum_{k=1}^{K} w_k\,\mathcal{M}\left(\frac{\mathbf{n}_s}{\|\mathbf{n}_s\|}\,\middle|\,\boldsymbol{\mu}_k, \kappa_k\right) \tag{5}$$

1.2. Variable to Factor Messages

Next, we present the variable-to-factor messages for all variables in the MRF. These messages are the product of the incoming messages to the variable except the message from the destination variable (see Eq. 2) and given as follows:

Message from occupancy variable to the appearance ray potential:

$$\mu_{o_i^r \to \psi_r^{\mathbf{a}}}(o_i^r) = \mu_{\varphi_i^o \to o_i}(o_i^r) \times \mu_{\psi_r^d \to o_i^r}(o_i^r) \tag{6}$$

Message from occupancy variable to the depth ray potential:

$$\mu_{o_i^r \to \psi_r^d}(o_i^r) = \mu_{\varphi_i^o \to o_i}(o_i^r) \times \mu_{\psi_r^{\mathbf{a}} \to o_i^r}(o_i^r) \tag{7}$$

Message from depth variable to the depth ray potential:

$$\mu_{d_r \to \psi_r^d}(d_r) = \mu_{\psi_{sr}^p \to d_r}(d_r) \tag{8}$$

Message from depth variable to the plane depth potential:

$$\mu_{d_r \to \psi_{sr}^p}(d_r) = \mu_{\psi_r^d \to d_r}(d_r) \tag{9}$$

Message from planarity variable to the plane depth potential:

$$\mu_{p_s \to \psi_{sr}^p}(p_s) = \mu_{\varphi_s^p \to p_s}(p_s) \tag{10}$$

Message from plane parameter variable to the plane depth potential:

$$\mu_{\mathbf{n}_s \to \psi_{sr}^p}(\mathbf{n}_s) = \mu_{\varphi_s^{\mathbf{n}} \to \mathbf{n}_s}(\mathbf{n}_s) \tag{11}$$

1.3. Derivation for the depth ray potential messages

The depth d_r at pixel r is related to the voxel occupancies $\mathbf{o}_r$ along ray r via the depth ray potential ψ_r^d defined as follows,

$$\psi_r^d(\mathbf{o}_r, d_r) = \begin{cases} 1 & \text{if } d_r = \sum_{i=1}^{N_r} o_i^r \prod_{j<i} (1 - o_j^r)\, d_{ri} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where d_{ri} denotes the depth of voxel i along ray r. This potential establishes the connection between voxel and pixel space in our hybrid representation.

Message to the depth variable: We begin by deriving the factor-to-variable messages to the depth variable d_r. In the following equations, we drop the ray index r for notational clarity. The message reads as

$$\mu_{\psi^d \to d}(d = d_i) = \sum_{o_1} \sum_{o_2} ... \sum_{o_N} \psi^d(\mathbf{o}, d_i) \prod_{j=1}^{N} \mu(o_j) \tag{13}$$

where we have abbreviated the incoming messages to the occupancy variables as follows, $\mu(o_j) = \mu_{o_j \to \psi^d}(o_j)$. Note that naive computing of this message is intractable as the summation over the $\mathbf{o}$ variables requires $O(2^N)$ operations. However, the special algebraic form of the potential $\psi^d(\mathbf{o}, d)$ allows omputing this message in linear time as we demonstrate below.

We begin by expanding the summations over the occupancy variables as follows:

$$\begin{aligned} \mu_{\psi^d \to d}(d = d_i) = & \mu(o_1 = 1) \Bigg[\sum_{o_2} ... \sum_{o_N} \psi^d(o_1 = 1, o_2, ..., .o_N, d = d_i) \prod_{j=2}^{N} \mu(o_j) \Bigg] + \\ & \mu(o_1 = 0) \Bigg[\sum_{o_2} ... \sum_{o_N} \psi^d(o_1 = 0, o_2, ..., o_N, d = d_i) \prod_{j=2}^{N} \mu(o_j) \Bigg]. \end{aligned} \tag{14}$$

If we assume that $i \neq 1$, then $\psi^d(o_1 = 1, o_2, ...o_N, d_i) = 0$ and the upper term in the square brackets vanishes, simplifying the message to

$$\mu_{\psi^d \to d}(d = d_i) = \mu(o_1 = 0)\Bigg[\sum_{o_2} ... \sum_{o_N} \psi^d(o_1 = 0, o_2, ..., o_N, d = d_i) \prod_{j=2}^{N} \mu(o_j)\Bigg]. \tag{15}$$

This strategy is repeated until reaching the summation over the ith occupancy variable. At this point, the message reads

$$\mu_{\psi^d \to d}(d = d_i) = \Bigg[\prod_{j<i} \mu(o_j = 0)\Bigg]\Bigg[\sum_{o_i} ... \sum_{o_N} \psi^d(o_1 = 0, o_2 = 0, ..., o_{i-1} = 0, o_i, ..., o_N, d = d_i) \prod_{j=i}^{N} \mu(o_j)\Bigg] \tag{16}$$

$$\begin{aligned} &= \Bigg[\prod_{j<i} \mu(o_j = 0)\Bigg]\mu(o_i = 1)\Bigg[\sum_{o_{i+1}} ... \sum_{o_N} \overbrace{\psi^d(o_1 = 0, o_2 = 0, ..., o_i = 1, o_{i+1}, ..., o_N, d = d_i)}^{\text{evaluates to 1}} \prod_{j=i+1}^{N} \mu(o_j)\Bigg] + \\ &\Bigg[\prod_{j<i} \mu(o_j = 0)\Bigg]\mu(o_i = 0)\Bigg[\sum_{o_{i+1}} ... \sum_{o_N} \underbrace{\psi^d(o_1 = 0, o_2 = 0, ..., o_i = 0, o_{i+1}, ..., o_N, d = d_i)}_{\text{evaluates to 0}} \prod_{j=i+1}^{N} \mu(o_j)\Bigg]. \end{aligned} \tag{17}$$

Since the potential ψ^d_r in the bottom summand evaluates to 0, the entire bottom term vanishes and the message is simplified to

$$\mu_{\psi^d \to d}(d = d_i) = \Bigg[\prod_{j<i} \mu(o_j = 0)\Bigg]\mu(o_i = 1)\underbrace{\Bigg[\sum_{o_{i+1}} ... \sum_{o_N} \prod_{j=i+1}^{N} \mu(o_j)\Bigg]}_{\text{evaluates to 1}}. \tag{18}$$

As all incoming messages $\mu(o_j)$ sum to 1, the term inside the rightmost brackets evaluates to 1 and the message is further simplified to

$$\mu_{\psi^d \to d}(d = d_i) = \mu(o_i = 1) \prod_{j<i} \mu(o_j = 0) \tag{19}$$

Message to the occupancy variables: We begin deriving the positive message:

$$\mu_{\psi^d \to o_i}(o_i = 1) = \sum_{d=d_1}^{d_N} \mu(d) \sum_{o_1} ... \sum_{o_{i-1}} \sum_{o_{i+1}} ... \sum_{o_N} \psi^d(o_1, ..., o_i = 1, ..., o_N, d) \prod_{\substack{j=1 \\ j \neq i}}^{N} \mu(o_j) \tag{20}$$

Like above, we have abbreviated the incoming messages to the occupancy and depth variables using the shorthand notation $\mu(o_j) = \mu_{o_j \to \psi^d}(o_j)$ and $\mu(d) = \mu_{d \to \psi^d}(d)$. We expand the summation over the depth variable according to its relative depth wrt. voxel i:

$$\begin{aligned} \mu_{\psi^d \to o_i}(o_i = 1) = &\sum_{d_j=d_1}^{d_{i-1}} \mu(d_j) \underbrace{\Bigg[\sum_{o_1} ... \sum_{o_{i-1}} \sum_{o_{i+1}} ... \sum_{o_N} \psi^d(o_1, ..., o_i = 1, ..., o_N, d_j) \prod_{\substack{j=1 \\ j \neq i}}^{N} \mu(o_j)\Bigg]}_{(\square)} \\ &+ \mu(d_i) \underbrace{\Bigg[\sum_{o_1} ... \sum_{o_{i-1}} \sum_{o_{i+1}} ... \sum_{o_N} \psi^d(o_1, ..., o_i = 1, ..., o_N, d = d_i) \prod_{\substack{j=1 \\ j \neq i}}^{N} \mu(o_j)\Bigg]}_{(\triangle)} \\ &+ \sum_{d_j=d_{i+1}}^{d_N} \mu(d_j) \underbrace{\Bigg[\sum_{o_1} ... \sum_{o_{i-1}} \sum_{o_{i+1}} ... \sum_{o_N} \psi^d(o_1, ..., o_i = 1, ..., o_N, d_j) \prod_{\substack{j=1 \\ j \neq i}}^{N} \mu(o_j)\Bigg]}_{(\Diamond)}. \end{aligned} \tag{21}$$

We can compute $(\Box)$ using the same strategy used to derive the message to the depth variable. It can be verified that, for any $d = d_j$ such that $d_j \leq d_{i-1}$, we have

$$(\Box) = \mu(o_j = 1) \prod_{k<j} \mu(o_k = 0). \tag{22}$$

Evaluating $(\triangle)$

$$(\triangle) = \sum_{o_1} ... \sum_{o_{i-1}} \sum_{o_{i+1}} ... \sum_{o_N} \psi^d(o_1, ..., o_i = 1, ..., o_N, d = d_i) \prod_{\substack{j=1 \\ j \neq i}}^{N} \mu(o_j), \tag{23}$$

we realize that $\psi^d(o_1, ..., o_i = 1, ..., o_N, d = d_i)$ evaluates to 1 if $o_j = 0$ for all $j < i$, and 0 for all other cases. Thus we obtain

$$(\triangle) = \left[\prod_{j<i} \mu(o_j = 0)\right] \underbrace{\left[\sum_{o_{i+1}} ... \sum_{o_N} \prod_{j=i+1}^{N} \mu(o_j)\right]}_{\text{evaluates to 1}} = \prod_{j<i} \mu(o_j = 0). \tag{24}$$

Finally, $(\Diamond)$ evaluates to 0 since $\psi^d(o_1, ..., o_i = 1, ..., o_N, d) = 0$ for all $d > d_i$.
By combining all results above, we write the positive message equation as follows:

$$\mu_{\psi^d \to o_i}(o_i = 1) = \sum_{j=1}^{i-1} \mu(d_j) \left[\mu(o_j = 1) \prod_{k<j} \mu(o_k = 0)\right] + \mu(d_i) \left[\prod_{k<i} \mu(o_k = 0)\right] \tag{25}$$

The derivation for the negative case can be obtained by following similar arguments, yielding:

$$\mu_{\psi^d \to o_i}(o_i = 0) = \sum_{j=1}^{i-1} \mu(d_j)\, \mu(o_j = 1) \prod_{k<j} \mu(o_k = 0) + \sum_{j=i+1}^{N} \mu(d_j)\, \mu(o_j = 1) \prod_{\substack{k<j \\ k \neq i}} \mu(o_k = 0) \tag{26}$$

1.4. Particle Belief Propagation

We use particle belief propagation [1] in order to approximate the continuous message equations arising for the plane depth potential. In particular, we draw K particles, $\{\mathbf{n}_s^{(k)}\}_{k=1}^{K}$, from a proposal distribution $W_s(\mathbf{n})$. Using these particles, we approximate the integrals in the message equations $\mu_{\psi^p_{sr} \to p_s}(p_s)$ and $\mu_{\psi^p_{sr} \to d_r}(d_r)$ with a Monte carlo estimate. We present the equations for the exact and the approximated messages below. We abbreviate the incoming messages for brevity as follows: $\mu(d_r) = \mu_{d_r \to \psi^p_{sr}}(d_r)$, $\mu(\mathbf{n}_s) = \mu_{\mathbf{n}_s \to \psi^p_{sr}}(\mathbf{n}_s)$ and $\mu(p_s) = \mu_{p_s \to \psi^p_{sr}}(p_s)$. Note that the message approximations below are up to a proportionality factor. In practice, we renormalize all messages appropriately after approximation.

Message to the planarity variable:

$$\mu_{\psi^p_{sr} \to p_s}(p_s = 1) = \int_{\mathbf{n}_s} \sum_{d_r} \psi^p_{sr}(d_r, p_s = 1, \mathbf{n}_s) \mu(\mathbf{n}_s) \mu(d_r) \tag{27}$$

$$\approx \frac{1}{K} \sum_{k=1}^{K} \sum_{d_r} \psi^p_{sr}(d_r, p_s = 1, \mathbf{n}_s) \frac{\mu(\mathbf{n}_s^{(k)})}{W_s(\mathbf{n}_s^{(k)})} \mu(d_r) \tag{28}$$

$$\mu_{\psi^p_{sr} \to p_s}(p_s = 0) = \int_{\mathbf{n}_s} \sum_{d_r} \psi^p_{sr}(d_r, p_s = 0, \mathbf{n}_s)\, \mu(\mathbf{n}_s)\, \mu(d_r) = \int_{\mathbf{n}_s} \sum_{d_r} 1\, \mu(\mathbf{n}_s)\, \mu(d_r) \tag{29}$$

$$\approx \frac{1}{K} \sum_{k=1}^{K} \frac{\mu(\mathbf{n}_s^{(k)})}{W_s(\mathbf{n}_s^{(k)})} \tag{30}$$

Message to the depth variable:

$$\mu_{\psi^p_{sr}\to d_r}(d_r) = \sum_{p_s}\int_{\mathbf{n}_s} \psi^p_{sr}(d_r, p_s, \mathbf{n}_s)\mu(p_s)\mu(\mathbf{n}_s) \tag{31}$$

$$= \mu(p_s = 1)\int_{\mathbf{n}_s} \psi^p_{sr}(d_r, p_s = 1, \mathbf{n}_s)\mu(\mathbf{n}_s) + \mu(p_s = 0)\int_{\mathbf{n}_s} \psi^p_{sr}(d_r, p_s = 0, \mathbf{n}_s)\mu(\mathbf{n}_s) \tag{32}$$

$$\approx \mu(p_s = 1)\frac{1}{K}\sum_{k=1}^{K} \psi^p_{sr}(d_r, p_s = 1, \mathbf{n}_s^{(k)})\frac{\mu(\mathbf{n}_s^{(k)})}{W_s(\mathbf{n}_s^{(k)})} + \mu(p_s = 0)\,\frac{1}{K}\sum_{k=1}^{K} \frac{\mu(\mathbf{n}_s^{(k)})}{W_s(\mathbf{n}_s^{(k)})} \tag{33}$$

2. Inference Algorithm Pseudocode

In the following, we present detailed pseudocode for the inference procedure.

Algorithm 1 Inference algorithm

```
procedure INFERENCE
    Shuffle images.
    Initialize the depth ray potential messages, i.e. $\mu_{\psi^d_r \to o^r_i}(o^r_i)$, to uniform distributions.
    while not converged do
        for each image in the list do
            Perform message passing for the appearance ray potentials.
        REFINE OCTREE()
    while not converged do
        for each image in the list do
            Perform message passing for the appearance ray potentials.
            PLANARITY MESSAGE PASSING(image)
```

Algorithm 2 Message passing pseudocode for the planarity potentials

```
procedure PLANARITY MESSAGE PASSING(image)
    Compute the median of the (current) depth variable beliefs $p(d)$ for each pixel in the image.
    For each segment in the image, generate plane parameter samples $\{\mathbf{n}_s^{(k)}\}_{k=1}^K$ using the depth estimates as described
  in Section 4.2 in the submission.
    Compute the messages from each potential $\psi^p_{sr}$ to the plane parameter variables using the particles, i.e. evaluate
  $\mu_{\psi^p_{sr}\to \mathbf{n}_s}(\mathbf{n}_s^{(k)})$.
    Compute the belief of each plane parameter variable, i.e. $p(\mathbf{n}_s^{(k)}) \propto \prod_{r \in R_s} \mu_{\psi^p_{sr}\to \mathbf{n}_s}(\mathbf{n}_s^{(k)})$.
    Compute the message from each potential $\psi^p_{sr}$ to the binary planarity variables, i.e. $\mu_{\psi^p_{sr}\to p_s}(p_s)$, as described in
  Section 1.4.
    Compute the belief of each planarity variable, i.e. $p(p_s) \propto \prod_{r \in R_s} \mu_{\psi^p_{sr}\to p_s}(p_s)$
    Compute the message from each potential $\psi^p_{sr}$ to each depth variable, i.e. $\mu_{\psi^p_{sr}\to d_r}(d_r)$.
    Compute the message from each depth ray potential to the occupancy variables along respective rays, i.e.
  $\mu_{\psi^d_r \to o^r_i}(o^r_i)$.
    Update the occupancy belief of each voxel using the message $\mu_{\psi^d_r \to o^r_i}(o^r_i)$.
```

Algorithm 3 Octree refinement procedure

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

3. Bayes Optimal Depth Estimation

Our probabilistic model allows for computing depth maps which are optimal in terms of Bayes decision. Let us consider a single pixel / ray r. The optimal Bayes decision minimizes the expected loss as follows,

$$d_r^* = \underset{d_r}{\operatorname{argmin}} \ \mathbb{E}_{p(d_r')}[\Delta(d_r, d_r')]. \tag{34}$$

where $p(d_r')$ is the probability distribution of the depth variables according to the joint distribution, i.e. Eq. 2 in the main paper. If we consider the common ℓ_1-loss, $\Delta(d_r, d_r') = |d_r - d_r'|$, the minimizer to Eq. 34 is given by d_r^* where $p(d_r < d_r^*) = p(d_r \geq d_r^*) = 0.5$, i.e., the median of $p(d_r)$.

The marginal $p(d_r)$ can be computed by marginalizing all variables except for d_r from the joint distribution $p(\mathbf{o}, \mathbf{a}, \mathbf{d}, \mathbf{p}, \mathbf{n})$. While this computation is intractable in general, our inference procedure based on sum-product belief propagation yields an approximation to the marginal. In particular, the belief of the variable, i.e., the product of all incoming messages to the variable, approximates the marginal distribution of the variable as

$$p(d_r) \approx \mu_{\psi_r^d \to d_r}(d_r) \times \mu_{\psi_{sr}^p \to d_r}(d_r) \tag{35}$$

where $\mu_{\psi_r^d \to d_r}(d_r)$ and $\mu_{\psi_{sr}^p \to d_r}(d_r)$ are specified above and in the main paper, respectively.

4. Pairwise Smoothness Weights

As discussed in the submission, we compare our method to a baseline that incorporates pairwise smoothness constraints into [2], thereby encouraging adjacent voxels to take the same occupancy label.

4.1. Formulation

The joint distribution of this baseline is given by

$$p(\mathbf{o}, \mathbf{a}) = \frac{1}{Z} \underbrace{\prod_{i \in \mathcal{X}} \varphi_i^o(o_i) \prod_{r \in \mathcal{R}} \psi_r^{\mathbf{a}}(\mathbf{o}_r, \mathbf{a}_r)}_{\text{joint distribution of [2]}} \underbrace{\prod_{(i,j) \in \mathcal{N}} \phi(o_i, o_j)}_{\text{pairwise smoothness terms}} \tag{36}$$

where $\mathcal{N}$ is the set of adjacent voxel pairs in the grid. We consider a 6-neighborhood in the voxel grid, i.e., up, down, left, right, front and back neighbors. The pairwise potential ϕ is defined by a Potts model

$$\phi(o_i, o_j) = \begin{cases} \gamma & \text{if } o_i = o_j \\ 1 & \text{if } o_i \neq o_j \end{cases} \tag{37}$$

which encourages neighboring voxels to take on the same label, i.e., $\gamma > 1$. The factor-to-variable message equations for the pairwise smoothness potential ϕ are given by

$$\mu_{\phi \to o_i}(o_i) = \sum_{o_j} \phi(o_i, o_j) \mu_{o_j \to \phi}(o_j) \tag{38}$$

$$\mu_{\phi \to o_i}(o_i = 1) = \gamma \, \mu_{o_j \to \phi}(o_j = 1) + \mu_{o_j \to \phi}(o_j = 0) \tag{39}$$

$$\mu_{\phi \to o_i}(o_i = 0) = \mu_{o_j \to \phi}(o_j = 1) + \gamma \, \mu_{o_j \to \phi}(o_j = 0) \tag{40}$$

4.2. Experiments

We now present experiments to study the effect of this simple pairwise smoothness prior by varying the parameter γ on the CAPITOL dataset. We run the inference procedure for various values of γ larger than and equal to 1. Note that for $\gamma = 1$ the pairwise potential becomes inactive and the algorithm reverts to the formulation of [2]. Fig. 2 presents the average precision curves for [2], which we refer to as "SP", [2] with pairwise terms, which we refer to as "SP+pairwise", as well as our algorithm. The average accuracy is computed as the area under the curve for the accuracy plots we used in the submission. Thus, higher numbers are better. It can be observed that the "SP+pairwise" algorithm's accuracy initially increases with γ. This result is intuitive and suggests that the pairwise smoothness potentials are helping to obtain better accuracy, mostly due to denser and smoother results. However, after a certain threshold, i.e. $\gamma \approx 2$, the accuracy starts to decrease and continues to decrease for larger values of γ. This is due to the shrinking bias inherent to the Potts model which removes more and more of the surfaces from the reconstruction. Compared to the best accuracy the "SP+pairwise" model achieves, our algorithm is significantly better as shown in Fig. 2. For all the experiments in the submission for the "SP+pairwise" algorithm, we used $\gamma = 1.75$ that yielded the best quantitative results.

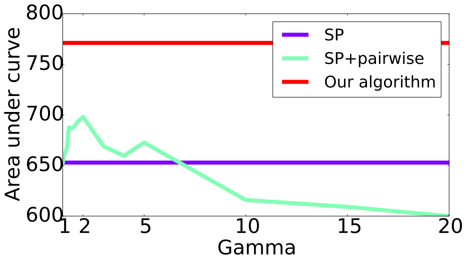
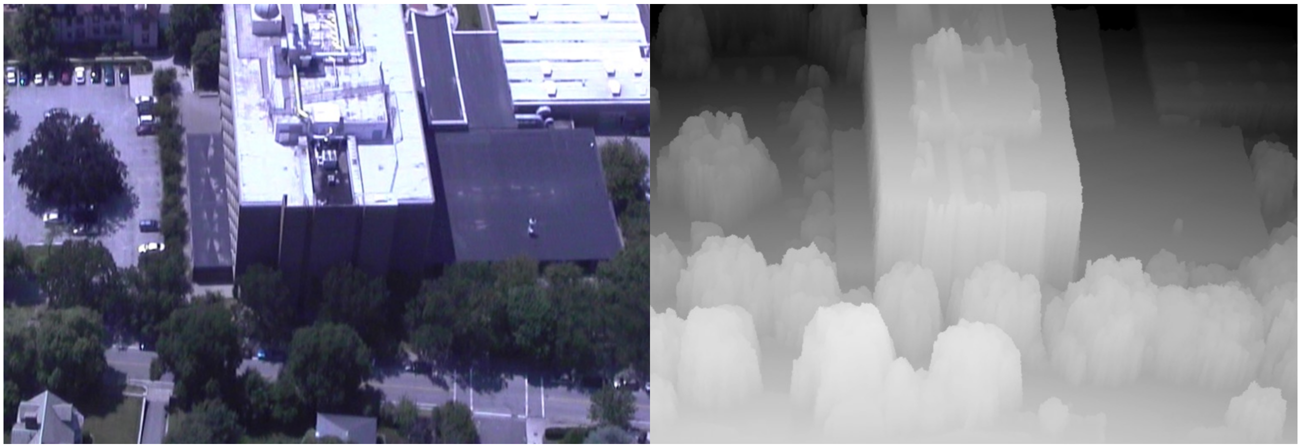


Figure 2: Average precision plots for the CAPITOL dataset for the sum-product algorithm without spatial smoothness terms [2], with pairwise smoothness terms, and our algorithm.

5. Evaluation for BARUS&HOLLEY Dataset Excluding the Trees and Vegetation

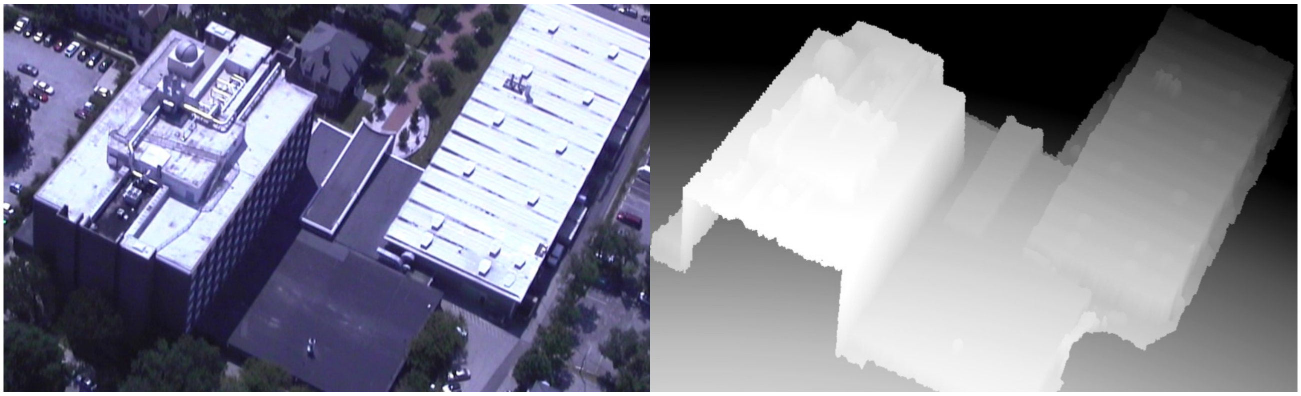


(a) Reference photo

(b) Ground truth depth map rendering

Figure 3: The ground truth generated by Ulusoy et al. [2] for the BARUS&HOLLEY dataset is not very accurate around the trees and other vegetation. Note the extrusion of the tree tops onto the ground.

As it was mentioned in the submission, the ground truth for the BARUS&HOLLEY dataset is not very accurate around the trees and other vegetation. Ulusoy et al. generated this ground truth by extruding the LIDAR point cloud onto the ground plane, assuming surfaces in the scene are mostly non-concave [2]. Although this assumption holds for buildings, roads, etc., it is violated for trees as shown in Fig. 3 and leads to incorrect ground truth around such tree regions.



(a) Reference photo

(b) Depth map rendering of the ground truth excluding the trees

Figure 4: The clipped ground truth for BARUS&HOLLEY dataset. We removed one side of the building from the ground truth mesh because it was being occluded by the trees.

In the BARUS&HOLLEY dataset, trees and vegetation occupy a large enough portion of the scene to affect the quantitative evaluation. In order to reduce the effects of this incorrect evaluation, we crop the ground truth mesh provided by [2] and remove the trees and vegetation as much as possible. The cropped mesh is shown in Fig. 4. We repeat the quantitative evaluation using this new ground truth mesh. The accuracy plots are shown in Fig. 5.

We also visualize the errors in Fig. 6. Note that algorithms PM and LC make large mistakes around the textureless black rooftop next to the building as shown in Fig. 9b+9c+9d. The sum-product result shown in Fig. 9e is much better on the rooftop but still contains errors. In particular, a slightly reflective region of the rooftop contains a hole, leading to large errors. We present enlarged images of the results on this region in Fig. 7. The baseline that incorporates pairwise smoothness potentials into the sum-product formulation, which we refer to as “SP+pairwise”, produces less noisy and smoother results as can be seen in Fig. 9f and Fig. 7f. However, Fig. 7f shows that the pairwise terms does not help with filling the large hole on the rooftop. Instead, our result, displayed in Fig. 7g, shows that our algorithm is able to regularize the entire roof structure and reconstruct the true roof geometry. Our algorithm accomplishes this regularization by propagating information from the two gray beams on the rooftop, where there is little ambiguity due to the sharp edge features, into the middle of the roof, where there is reconstruction ambiguity due to the reflective surface. Our overall result shown in Fig. 6g contains less error throughout the entire scene compared to the sum-product result as well as the baseline with pairwise terms. Fig. 5 confirms that our results are also quantitatively better than all other methods.

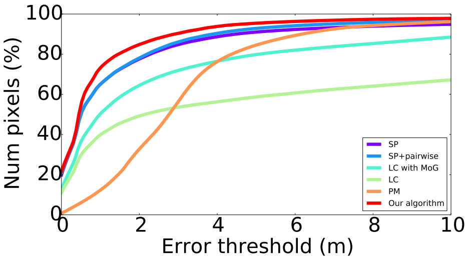
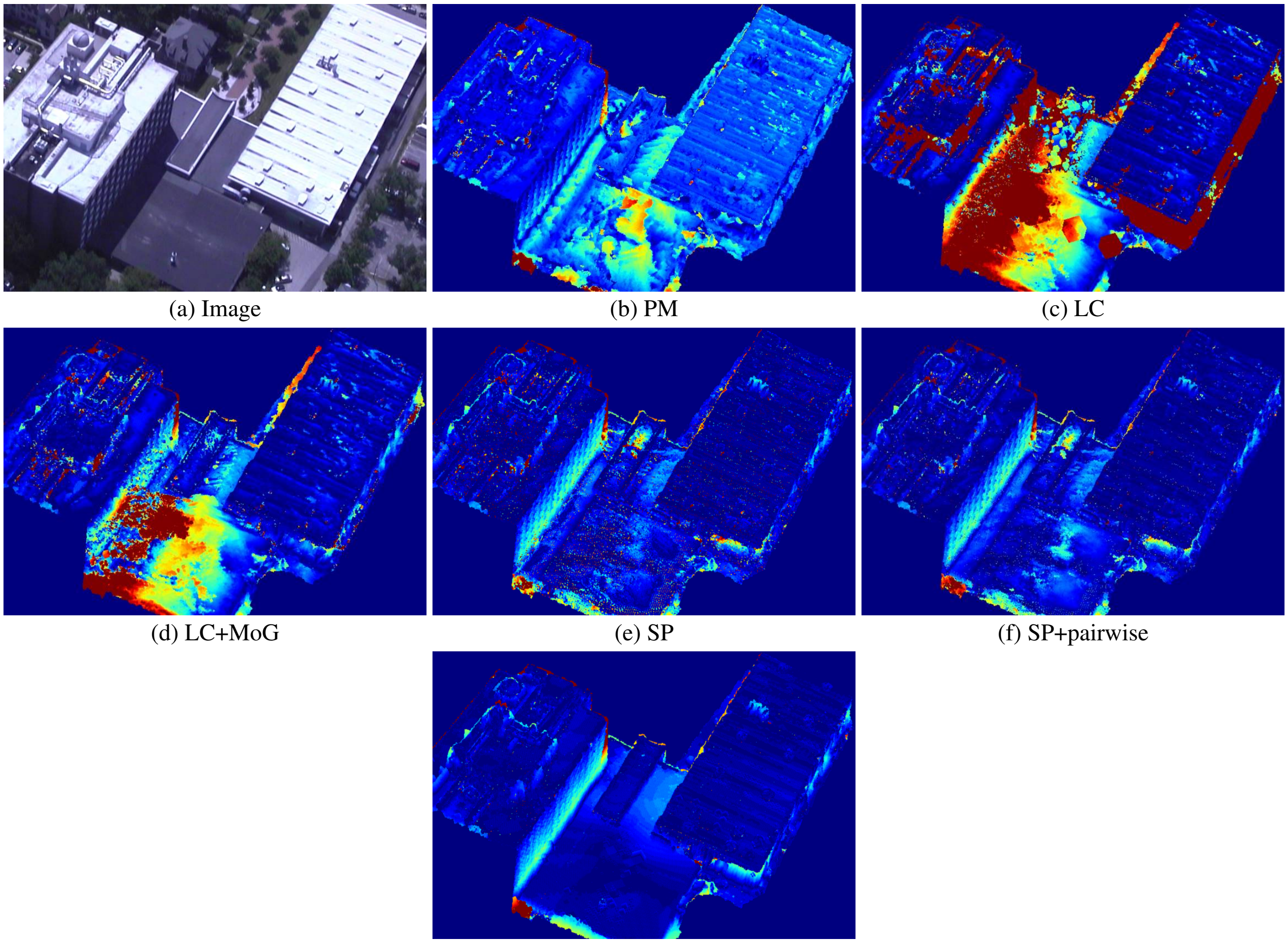


Figure 5: Accuracy plots for the BARUS&HOLLEY dataset using the new ground truth mesh shown in Fig. 4.



(a) Image

(b) PM

(c) LC

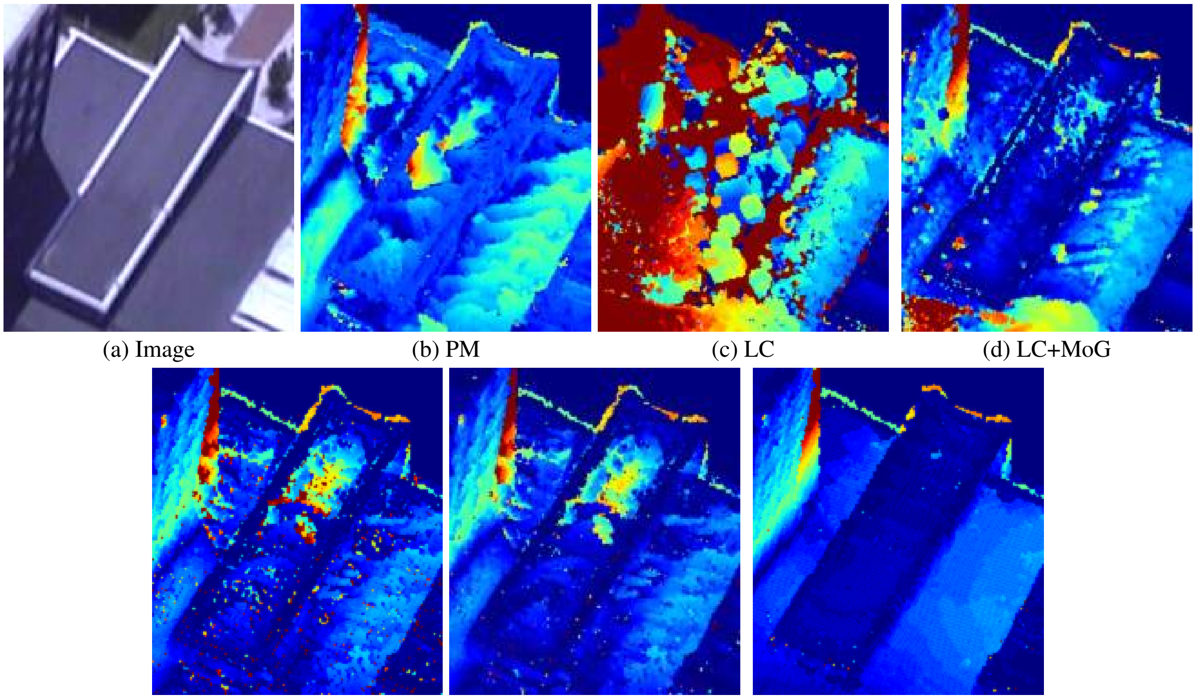
(d) LC+MoG

(e) SP

(f) SP+pairwise

(g) Ours

Figure 6: Visualization of depth map prediction error for each algorithm using the clipped ground truth which excludes the trees and vegetation.



(a) Image (b) PM (c) LC (d) LC+MoG

(e) SP (f) SP+pairwise (g) Ours

Figure 7: Visualization of depth map prediction error for each algorithm using the clipped ground truth that excludes the trees and vegetation.

6. Additional Experiments using a Uniform Prior over Plane Normals

In this work, we utilize a Manhattan world prior on the plane orientations. We formulate this prior using a mixture of von-Mises Fisher distributions:

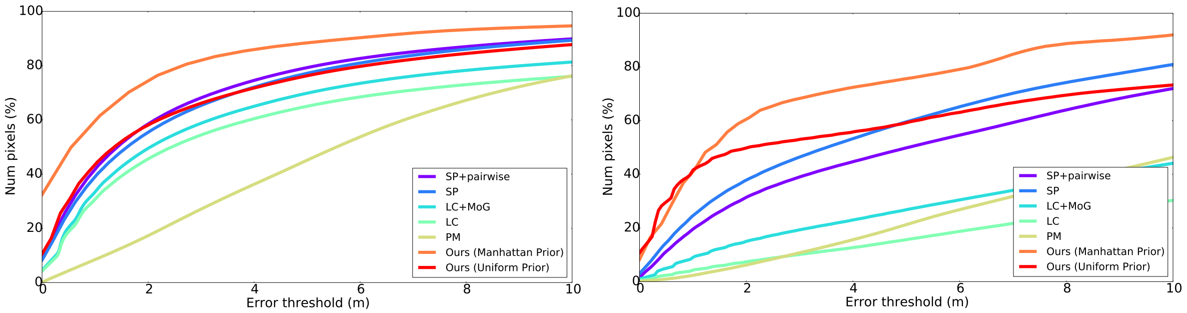
$$\varphi_s^{\mathbf{n}}(\mathbf{n}_s) = \sum_{k=1}^{K} w_k \, \mathcal{M}\left(\frac{\mathbf{n}_s}{\|\mathbf{n}_s\|} \,\middle|\, \boldsymbol{\mu}_k, \kappa_k\right) \tag{41}$$

where $\mathcal{M}(\cdot \mid \boldsymbol{\mu}, \kappa)$ denotes the von Mises-Fisher distribution with parameters $\boldsymbol{\mu}$ and κ. This choice of prior is reasonable for the urban scenes we consider in this work because most surfaces are aligned with the X, Y or Z directions. In order to demonstrate the effectiveness of our prior, we include a sample experiment that instead of using the Manhattan world prior, utilizes a *uniform prior* over the plane orientations. For this experiment, we simply set the $\varphi_s^{\mathbf{n}}(\mathbf{n}_s)$ to a uniform distribution.

We present accuracy plots for the CAPITOL dataset in Fig. 8. It can be observed that our algorithm with the uniform prior performs worse than using the Manhattan world prior. However, even when using this uniform prior, our algorithm outperforms most other algorithms, and achieves comparable results to the sum-product, “SP”, result.

We visualize the errors in depth prediction in Fig. 9. It can be observed that the algorithm with the uniform prior over plane orientations improves upon the the sum-product result, i.e. “SP”, which doesn’t contain any spatial priors, and also significantly outperforms the sum-product algorithm with the pairwise smoothness potentials, i.e. “SP+pairwise”.

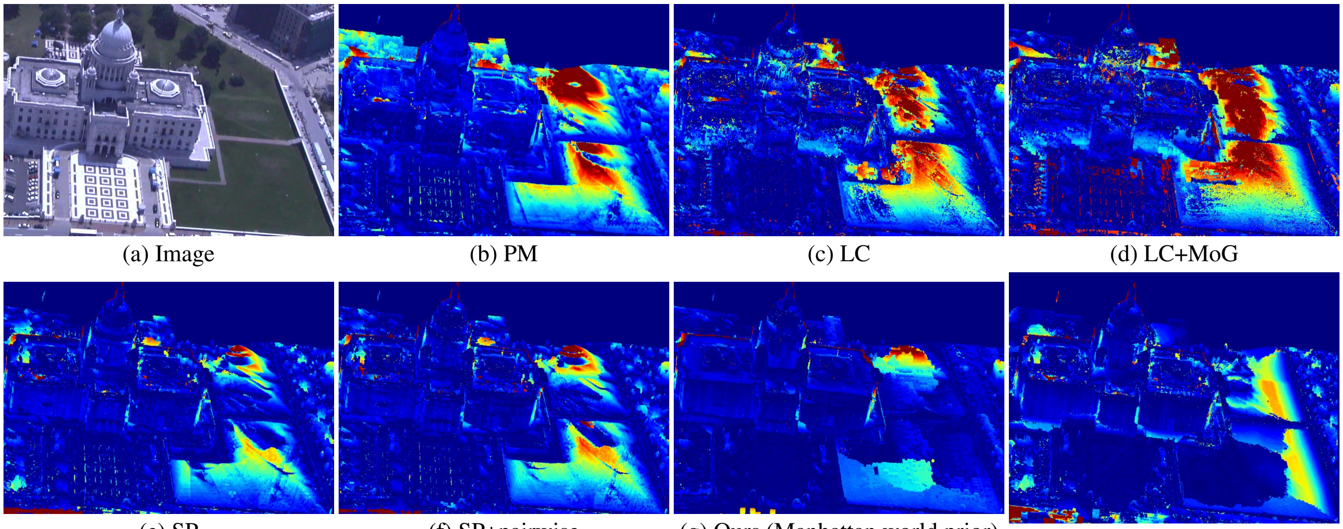
Fig. 9h shows that most of the errors are concentrated on the right side of the grass region. Note that the overall grass region contains very few features and therefore contains a high degree of reconstruction ambiguity. Our analysis indicates that for such ambiguous regions, plane proposals computed by RANSAC are not always reliable. In addition to the RANSAC proposals, our algorithm utilizes the Manhattan world prior to generate further plane proposals that are aligned with the X, Y or Z directions, as discussed in Section 4.2 in the submitted document. As Fig. 9g shows, this strategy is successful in reconstructing the correct geometry over the entire grass region. Instead, the algorithm with the uniform prior relies only on the generic RANSAC plane proposals and has difficulty reconstructing the full grass region.



(a) CAPITOL (all)

(b) CAPITOL (grass region)

Figure 8: Accuracy plots for the CAPITOL dataset using a *uniform* prior over the plane orientations.



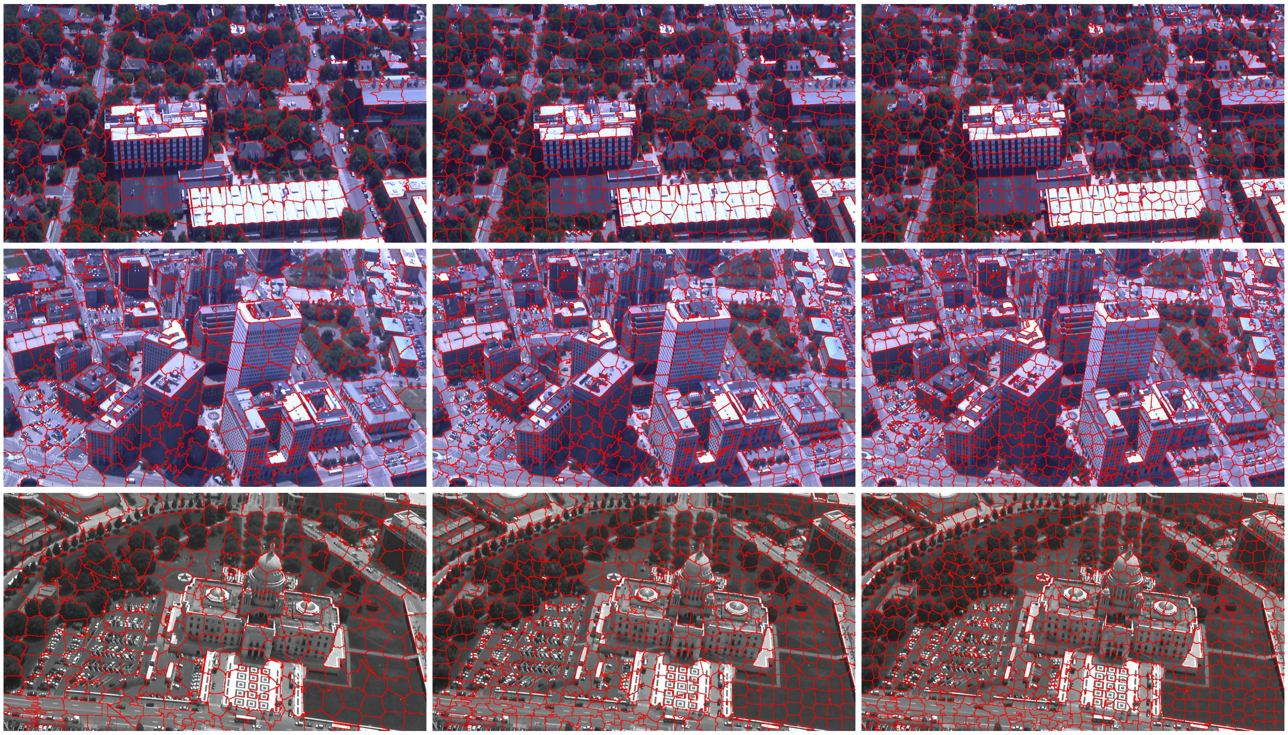
(a) Image (b) PM (c) LC (d) LC+MoG

(e) SP (f) SP+pairwise (g) Ours (Manhattan world prior) (h) Ours (Uniform prior)

Figure 9: Visualization of the depth prediction errors for the CAPITOL dataset.

7. Varying Segmentation Granularities

For the experiments presented in the submitted document, we set the parameters of the superpixelization algorithm [3] such that it generates roughly 500 segments. We empirically found this to be a reasonable tradoff between over- and under-segmentation of the images. We have also generated segmentations that roughly contain 200 and 750 segments using the same superpixelization algorithm [3]. Example segmentations can be seen in Fig. 10. Our initial experiments suggest that all three segmentation granularities yield similar performance.



(a) 250 superpixels

(b) 500 superpixels

(c) 750 superpixels

Figure 10: Superpixel segmentation results with varying number of segments.

References

[1] A. Ihler and D. McAllester. Particle belief propagation. *AISTATS*, 2009. 5

[2] A. O. Ulusoy, A. Geiger, and M. J. Black. Towards probabilistic volumetric reconstruction using ray potentials. In *3DV*, 2015. 7, 8, 9

[3] K. Yamaguchi, D. McAllester, and R. Urtasun. Robust monocular epipolar flow estimation. In *CVPR*, 2013. 12