Monocular Depth Estimation via Deep Structured Models with Ordinal Constraints

Daniel Ron* CMU / Amazon dron@alum.mit.edu

Kun Duan Snap Inc. kun.duan@snap.com

Chongyang Ma Snap Inc. cma@snap.com

Ning Xu Snap Inc. ning.xu@snap.com

Shenlong Wang University of Toronto slwang@cs.toronto.edu

Sumant Hanumante Snap Inc. shanumante@snap.com

Dhritiman Sagar Snap Inc. dman@snap.com

Abstract

User interaction provides useful information for solving challenging computer vision problems in practice. In this paper, we show that a very limited number of user clicks could greatly boost monocular depth estimation performance and overcome monocular ambiguities. We formulate this task as a deep structured model, in which the structured pixel-wise depth estimation has ordinal constraints introduced by user clicks. We show that the inference of the proposed model could be efficiently solved through a feed-forward network. We demonstrate the effectiveness of the proposed model on NYU Depth V2 *and* Stanford 2D-3D *datasets. On both datasets, we achieve state-of-the-art performance when encoding user interaction into our deep models.*

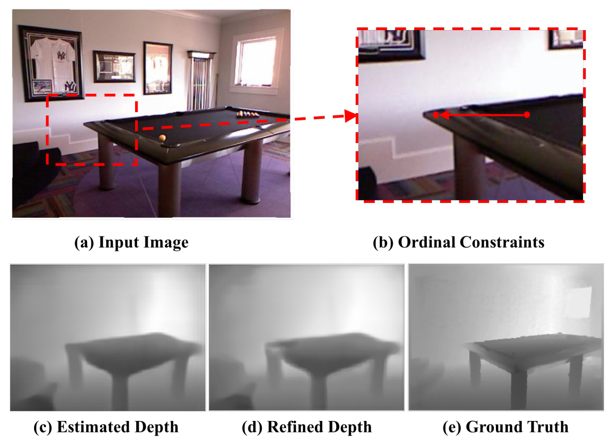


Figure 1: Overview of our work: we estimate depth from single input RGB image (a) while preserving the ordinal constraints provided by relative depth orders in the form of a click pair (b). Such ordinal constraints are modeled by our ADMM depth refinement network and the base network output (c) is passed through ADMM network modules to generate the refined depth prediction (d). We show the ground truth depth (e) as comparison.

1. Introduction

Depth prediction from a monocular RGB image is very useful in applications such as augmented reality, image re-focusing, face parsing, etc., where coarse depth information is required but no additional source of signals exists (e.g., depth sensor and camera motion). However, estimating depth from a single image is inherently an ill-posed problem, due to the ambiguous mapping between the 3D geometry of shapes and their appearance. State-of-the-art methods ([7, 14, 10, 15, 21, 34, 9, 33]) achieve promising results, estimating the depth from monocular images through deep learning algorithms. However, the inherent ambiguities of monocular depth estimation make predicting detailed geometry difficult. Even in trivial cases, the insufficient training data can lead to prediction failure.

People have demonstrated that it is possible to use depth prediction models for post-capture applications (e.g. refocus, background blur) that do not require real-time processing. In this work, we help the system correct and, consequently, improve the depth estimation result by leveraging user interactions. Such a semi-automatic system can be effective because simple user interactions like pairs of clicks can provide useful prior information about ambiguous regions.

The computer vision community has a long history of using small amounts of user interactions to remove ambiguities in computer vision tasks. For example, past research has investigated interactive image matting [26], intrinsic image decomposition [3] and image foreground segmentation [23, 35]. However, there is no existing work that investigates how user interaction could improve

*This work was conducted while Daniel Ron was an intern at Snap Inc.

monocular depth estimation at test time.

Based on this key observation, we propose a novel system for interactively predicting depth from a single RGB image. We formulate the task as a constrained quadratic programming problem with user interactions modeled as ordinal constraints. Figure 1 illustrates the idea of such constrained inference with ordinal relations. To solve the problem, we derive an iterative ADMM solver which can be implemented as standard neural network modules. We augment deep neural networks with this structured model to refine depth prediction results based on additional hard constraints.

In summary, our contributions are threefold. First, we propose including user interaction as ordinal constraints for monocular depth estimation tasks. Second, we derive and implement a novel deep structured model that handles such hard constraints. Lastly, we evaluate our proposed monocular depth estimation neural network on state-of-the-art datasets and show that our novel strategy outperforms baseline methods, both quantitatively and qualitatively.

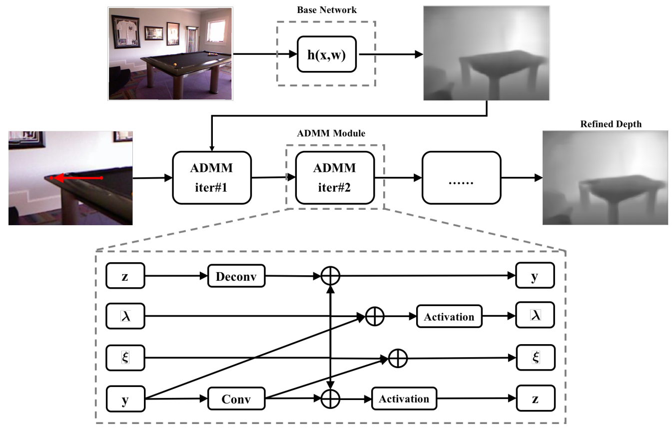


Figure 2: Our end-to-end network for monocular depth estimation with ordinal constraints. Such hard constraints are embedded into ADMM modules as network forward passes. Each ADMM module represents one iteration of ADMM update step for $\mathbf{y}$, $\mathbf{z}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\xi}$. Base network can be any network that generates possibly ambiguous monocular depths. We choose FCRN [15] as our base network in all our experiments.

2. Related Work

Training fully automatic machine learning systems for monocular depth estimation is a very challenging task. It requires curating large scale representative training datasets as well as sophisticated models, such as very deep neural networks [25, 13, 28]. Recent approaches on end-to-end deep neural networks for monocular depth estimation try to leverage extra signals (segmentation, surface normals, depth gradient or camera aperture [18, 31, 7, 16, 5, 21, 27]) to help solve such ambiguities. Although these methods explicitly explore the information hidden inside RGB images or depth maps, the depth inference problem itself still remains ill-posed. As discussed below, our work is related to a few important topics.

Deep structured models. A Markov Random Field (MRF) captures structured information in label space through *unary* potentials (or data terms) and *pairwise* potentials (or smoothness terms). MRF models are flexible to model complex relations with Markov or higher order relations. Efficient inference exists if potential functions are of particular forms, and training can be done discriminatively [30, 8]. Recent work on deep neural nets shows that such MRF inference can be rewritten as neural net layer operations and therefore the parameters can be learned using standard backprop gradient descent [11, 32, 36, 22]. For example, [32] proposes proximal net which rewrites proximal gradient descent rules as a combination of convolution, deconvolution, and non-linear activations. Our approach for modeling user interactions is related to this method because we formulate such guidance using ordinal pairs in an MRF. We use a similar primal-dual method and the *Alternating Direction Method of Multipliers* (ADMM) framework [36] to solve the depth prediction problem.

Human-in-the-loop. Our work is related to *active learning* or *human-in-the-loop* [6]. In such a framework, models are trained in an iterative manner by integrating user feedback as part of the loss function. In our work, we model user click pairs as ordinal constraints and use them to improve depth estimation models learned from single view RGBD training data. Our method can be formulated into active learning framework with iteratively added user click pairs based on the incorrect depths that current model predicts on a validation set. We simulate user click pairs from ground truth depths in our experiments and leave the exploration of active learning framework as worthwhile future explorations.

Modeling ordinal relations. Past research shows that humans are good at estimating relative depth order between points rather than metric depth values [29]. [4] studies the problem by designing a ranking loss function with such relative depth orderings, and training their model on a large set of weakly annotated web images. [37] solves a similar problem but uses constrained quadratic programming on superpixel segmentations. Both of these methods rely on ordinal click pairs only at the training time, while ours aims at adding such ordinal constraints at inference time via an MRF, as well as learning the parameters through neural network backpropagation. We design our own neural network modules where each forward pass implements one iteration of the corresponding ADMM update rules. This allows us to train the network in an end-to-end fashion.

Modeling single view priors. There are other types of prior information that can also help improve monocular depth estimation. For example, segmentation cues have been used to refine depth prediction [31, 17]. Such segmentation priors provide semantic boundary information in the scene and are particularly useful in preserving depth discontinuity at segmentation boundaries. In addition, some recent methods propose using 3D signals such as surface normals [7, 5] and depth gradients [16] to train the model for depth estimation. Our proposed method allows the flexibility to incorporate such segmentation-aware or gradient-aware priors in the form of high order potentials. Extra priors can always augment end-to-end neural nets but do not fit into the scope of this paper.

Our work is most relevant to [4] which trains an end-to-end neural network on annotations of relative depth. Their approach is able to train monocular depth estimation models on a much larger scale of data even when accurate ground truth depths are not available. Our proposed method differs from using annotations of relative depth during the training process. Instead, we encode relative depth orders as hard constraints at the inference time. We show that such interactions provide useful guidance at ambiguous pixels and significantly improve the depth prediction quality.

3. Approach

The user provides pairs of clicks which specify relative orders between pairs of pixels in depth direction. We consider such user guidance as pairwise ordinal constraints on inferred depth estimations. We use a similar approach as [37] and obtain such user guidance by sampling from ground truth depth that simulate human perception. We first describe our problem formulation as quadratic programming with linear constraints (Section 3.1), and then explain the details of each step in our proposed ADMM solver (Section 3.2). We show how to convert our iterative algorithm into computation flow with neural network operations (Section 3.3).

3.1. Objective Function

Let N be the total number of pixels in an image, $\mathbf{x}$ and $\mathbf{y}$ are the vector representations for the input image and refined depth we want to solve for. We assume our refined depth values $\mathbf{y}$ are bounded within a range $[0, D]$. Given M pairs of ordinal constraints from user guidance, our objective function for optimizing $\mathbf{y}$ can be written as:

$$\mathbf{y}^* = \underset{\mathbf{y}}{\operatorname{argmin}}\, f_u(\mathbf{y}, \mathbf{x}) + \sum_{\alpha} f_p(\mathbf{y}_\alpha, \mathbf{x}) \tag{1}$$

s.t.

$$\mathbf{A}\mathbf{y} \leq \mathbf{B}$$

where $\mathbf{A} = \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \\ \mathbf{P} \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} \mathbf{0} \\ D\mathbf{1} \\ \mathbf{0} \end{bmatrix}$, $\mathbf{I}$ is the identity matrix, $\mathbf{0}$ and $\mathbf{1}$ are vectors of all 0s and 1s. $f_u(\mathbf{y}, \mathbf{x})$ is the unary potential encoding the prediction from a base deep neural network. $f_p(\mathbf{y}_\alpha, \mathbf{x})$ is the high-order potential encoding spatial relationship between neighboring pixels. $\mathbf{A}\mathbf{y} \leq \mathbf{B}$ encodes the hard constraints for ordinal relations. The first two parts in $\mathbf{A}$ and $\mathbf{B}$ ensure that the refined depth outputs are within the valid range $[0, D]$. $\mathbf{P}$ is a $M \times N$ matrix encoding M different ordinal constraints. We use $P_{kj} = 1$ and $P_{kj'} = -1$ if (j, j') is an ordinal pair where $k \leq M$.

First we assume our unary potentials f_u are of the form $f_u(\mathbf{y}, \mathbf{x}; \mathbf{w}) = \frac{1}{2}\|\mathbf{y} - h(\mathbf{x}; \mathbf{w})\|_2^2$ which measures the squared L2 distance between $\mathbf{y}$ and $h(\mathbf{x}; \mathbf{w})$. In our case of estimating depths, $h(\mathbf{x}; \mathbf{w})$ indicates the output from a base depth prediction network (e.g. Eigen [7] or FCRN [15]) parameterized by the network weights $\mathbf{w}$. Minimizing the unary terms is equivalent to minimizing the mean squared error between refined depths and base network outputs.

We assume our high-order potentials f_p to be of the form $f_p(\mathbf{y}_\alpha, \mathbf{x}; \mathbf{w}) = h_\alpha(\mathbf{x}; \mathbf{w}) g_\alpha(\mathbf{W}_\alpha \mathbf{y})$. Here $\mathbf{W}_\alpha$ denotes a transformation matrix for a filtering operation, and $h_\alpha(\mathbf{x}; \mathbf{w})$ provides per-pixel guidance information that places stronger local smoothness for pixels on low-frequency edges (similar to bilateral filter [20] or guided filter [12]). In our implementation, we assume $h_\alpha(\mathbf{x}; \mathbf{w})$ is constant for all the pixels since our goal is to demonstrate improvement from ordinal constraints. We designate edge-aware or segmentation-aware priors as future work.

3.2. Inference with Deep Structured Network

To solve for refined depth values $\mathbf{y}$, we apply the ADMM algorithm due to its capability of handling non-differentiable objectives and hard constraints, as well as its fast convergence. We introduce auxiliary variables $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_\mathcal{A}\}$ and rewrite the above formulation in Equation 1 as:

$$\mathbf{y}^* = \underset{\mathbf{y}}{\operatorname{argmin}}\, \frac{1}{2}\|\mathbf{y} - h(\mathbf{x}; \mathbf{w})\|_2^2 + \sum_{\alpha} h_\alpha(\mathbf{x}; \mathbf{w}) g_\alpha(\mathbf{z}_\alpha) \tag{2}$$

s.t.

$$\mathbf{A}\mathbf{y} \leq \mathbf{B}$$
$$\mathbf{W}_\alpha \mathbf{y} = \mathbf{z}_\alpha, \;\; \mathbf{z}_\alpha \in \mathbf{z}$$

The *augmented Lagrangian* of the original objective function can then be written as:

$$\begin{aligned} L_\rho(\mathbf{x}, \mathbf{y}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\xi}) = {} & \frac{1}{2}\|\mathbf{y} - h(\mathbf{x}; \mathbf{w})\|_2^2 + \sum_{\alpha} h_\alpha(\mathbf{x}; \mathbf{w}) g_\alpha(\mathbf{z}_\alpha) \\ & + \sum_{\alpha} \frac{\rho_\alpha}{2}\|\mathbf{W}_\alpha \mathbf{y} - \mathbf{z}_\alpha\|_2^2 + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{y} - \mathbf{B}) \\ & + \sum_{\alpha} \boldsymbol{\xi}_\alpha^T(\mathbf{W}_\alpha \mathbf{y} - \mathbf{z}_\alpha) \end{aligned}$$

where ρ_α is a constant penalty hyperparameter and $\boldsymbol{\lambda}, \boldsymbol{\xi}$ are Lagrange multipliers with $\boldsymbol{\lambda} \geq 0$. We iteratively solve for variables $\mathbf{y}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\xi}$ by alternating between the following subproblems.

Solve for refined depth $\mathbf{y}$. We calculate the derivative of Lagrangian function with respect to $\mathbf{y}$ to obtain its update rule:

$$\begin{aligned}\tilde{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y} - h(\mathbf{x}; \mathbf{w})\|_2^2 + \sum_\alpha \frac{\rho_\alpha}{2}\|\mathbf{W}_\alpha \mathbf{y} - \mathbf{z}_\alpha\|_2^2 \\ + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{y} - \mathbf{B}) + \sum_\alpha \boldsymbol{\xi}_\alpha^T(\mathbf{W}_\alpha \mathbf{y} - \mathbf{z}_\alpha) \\ = (\mathbf{I} + \sum_\alpha \rho_\alpha \mathbf{W}_\alpha^T \mathbf{W}_\alpha)^{-1}\Big(\boldsymbol{h}(\mathbf{x}, \mathbf{w}) - \mathbf{A}^T\boldsymbol{\lambda} \\ + \sum_\alpha \mathbf{W}_\alpha^T(\rho_\alpha \mathbf{z}_\alpha - \boldsymbol{\xi}_\alpha)\Big)\end{aligned}$$

Intuitively, this step uses the term $\mathbf{A}^T\boldsymbol{\lambda}$ to encode the ordinal constraints and adjust the outputs from base network. The depths are refined iteratively in a forward pass through ADMM network modules.

Solve for auxiliary variables $\mathbf{z}$. Let $g_\alpha(\cdot) = \|\cdot\|_1$ be the L1 smoothness priors on $\mathbf{y}$ and $S(a, b)$ be the *soft thresholding* function. We solve a *Lasso* problem to obtain the update rules for $\mathbf{z}$:

$$\begin{aligned}\tilde{\mathbf{z}} = \underset{\{\mathbf{z}_\alpha\}}{\operatorname{argmin}} h_\alpha(\mathbf{x}; \mathbf{w}) g_\alpha(\mathbf{z}_\alpha) + \frac{\rho_\alpha}{2}\|\mathbf{W}_\alpha \mathbf{y} - \mathbf{z}_\alpha\|_2^2 \\ + \sum_\alpha \boldsymbol{\xi}_\alpha^T(\mathbf{W}_\alpha \mathbf{y} - \mathbf{z}_\alpha)\end{aligned}$$

And for each $\mathbf{z}_\alpha$, we have:

$$\begin{aligned}\tilde{\mathbf{z}}_\alpha = \underset{\mathbf{z}_\alpha}{\operatorname{argmin}} h_\alpha(\mathbf{x}; \mathbf{w}) g_\alpha(\mathbf{z}_\alpha) + \frac{\rho_\alpha}{2}\|\mathbf{W}_\alpha \mathbf{y} - \mathbf{z}_\alpha\|_2^2 \\ + \boldsymbol{\xi}_\alpha^T(\mathbf{W}_\alpha \mathbf{y} - \mathbf{z}_\alpha) \\ = S(\mathbf{W}_\alpha \mathbf{y} + \frac{\boldsymbol{\xi}_\alpha}{\rho_\alpha}, \frac{h_\alpha(\mathbf{x}; \mathbf{w})}{\rho_\alpha})\end{aligned}$$

Solve for Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\xi}$. We can obtain a update rule for $\boldsymbol{\lambda}$ using gradient ascent as below:

$$\begin{aligned}\tilde{\boldsymbol{\lambda}} &= \max(\underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \boldsymbol{\lambda}(\mathbf{A}\mathbf{y} - \mathbf{B}), 0) \\ &= \max(\boldsymbol{\lambda} + \boldsymbol{\eta}(\mathbf{A}\mathbf{y} - \mathbf{B}), 0)\end{aligned}$$

Similarly for each $\boldsymbol{\xi}_\alpha$, we have gradient ascent update rule:

$$\begin{aligned}\tilde{\boldsymbol{\xi}_\alpha} &= \underset{\boldsymbol{\xi}_\alpha}{\operatorname{argmax}} \boldsymbol{\xi}_\alpha^T(\mathbf{W}_\alpha \mathbf{y}^{(n)} - \mathbf{z}_\alpha) \\ &= \boldsymbol{\xi}_\alpha + \boldsymbol{\tau}_\alpha(\mathbf{W}_\alpha \mathbf{y} - \mathbf{z}_\alpha)\end{aligned}$$

where $\boldsymbol{\eta}$ and $\boldsymbol{\tau}_\alpha$ are the hyperparameters denoting gradient update step sizes.

Our ADMM solver is iterative, and can be precisely implemented using a *recurrent neural network*. In practice, we do not share the weights in ADMM modules and we fix the number of iterations. This change allows us to implement our ADMM solver using a standard convolutional neural network with customized activation functions. A sketch of our end-to-end network with ADMM modules is shown in Figure 2.

3.3. Implementation Details

Here we discuss the implementation details of each step in our ADMM solver. We design an ADMM network module to run one iteration of the above update rules (Section 3.2). We learn the filters that encode the transformation $\mathbf{W}_\alpha$ using standard back propagation. $\mathbf{z}_\alpha$, $\boldsymbol{\xi}_\alpha$ and $\boldsymbol{\lambda}$ are data tensors and are initialized as zeros. We describe each layer in our ADMM module and their forward pass as below.

The first layer in our ADMM module is used to solve for refined depth $\mathbf{y}$. Calculating the numerator corresponds to applying a deconvolution (i.e. transposed convolution) step on each $\rho_\alpha \mathbf{z}_\alpha - \boldsymbol{\xi}_\alpha$ and taking the sum of results together. Calculating the denominator can be done by converting the deconvolution kernels to optical transfer functions [36] and taking the sum. It is possible to calculate the final output by first applying fast Fourier transform (FFT) on the numerator followed by an inverse FFT on the division result.

The second layer in our ADMM module is used to solve for auxiliary variables $\mathbf{z}$. This can be done with a convolution layer on $\mathbf{y}$, using the same filters shared with the deconvolution layer. The convolution layer output is passed through a non-linear activation layer that implements a standard soft thresholding function $S(a, b)$. In practice, we implement this soft thresholding function using two ReLU functions: $S(a, b) = \text{ReLU}(a-b) - \text{ReLU}(-a-b)$. We also do not force the convolution layer to share weights with the deconvolution layer in order to increase network capacity.

The third and fourth layers in our ADMM module correspond to gradient ascent steps that solve for Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\xi}$. These steps can be implemented as tensor subtraction and summation operations. We pass the updated result of $\boldsymbol{\lambda}$ after gradient ascent through an additional ReLU layer to satisfy the non-negative constraint on $\boldsymbol{\lambda}$.

In our experiments, we use five such modules for our ADMM network, which corresponds to running our solver for five iterations. Each ADMM module in our implementation contains 64 transformations $\mathbf{W}_\alpha$, i.e. we have 64 filters in each convolution and deconvolution layers. All the operations in our ADMM modules are differentiable and therefore the entire network (base net with ADMM network) can be learned end to end using gradient descent. We choose standard *mean squared error* (MSE) as the loss function in our experiments.

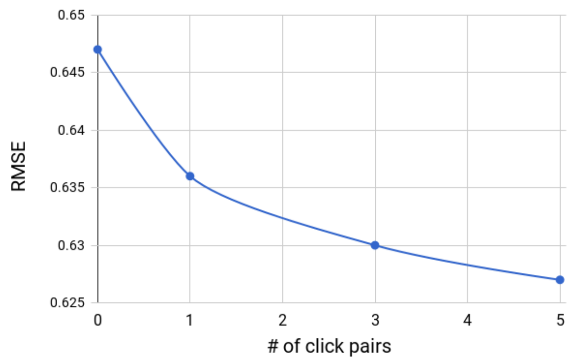


Figure 3: *Root Mean Squared Error* (RMSE) on NYU Depth V2 dataset with different numbers of click pairs $K = 0, 1, 3, 5$. For an image with total sampled click pairs fewer than K, we simply use all the click pairs available on that image. When $K = 0$, our network runs without any ordinal constraints and becomes equivalent to L1 regularized depth refinement network.

4. Experiments

4.1. Datasets

We evaluate our proposed approach on two state-of-the-art RGBD datasets: NYU Depth V2 [19] and Stanford 2D-3D-S [2]. We choose the state-of-art FCRN [15] as our base network for predicting initial depth value. On the NYU Depth V2 dataset, we finetune our end-to-end depth refinement network on 795 training images using the pre-trained FCRN network weights provided by the authors. Stanford 2D-3D-S is much larger and contains 70 496 RGBD images from five different indoor areas in total. We use a subset of area 2, area 4 and area 5 for training and a subset of area 1, area 3 and area 6 for testing [19]. Stanford 2D-3D-S also provides a binary mask for invalid raw depth pixels. We use such a binary mask to calculate loss only on valid pixels at the training phase. At the testing phase, we also mask out invalid pixels and only evaluate on valid pixels. Standard data augmentation including random left/right flip and random color distortion is applied when training our models on the two datasets.

On both datasets, we also use the same configuration for ADMM network modules. We set the constant penalty hyperparameter ρ to be 0.5 and set the filter size in both convolution and deconvolution layers to be 3. We initialize $\mathbf{z}$ and $\boldsymbol{\xi}$ as all zeros, and initialize $\boldsymbol{\lambda}$ as all ones. The gradient ascent step sizes for updating $\boldsymbol{\xi}$ and $\boldsymbol{\lambda}$ are set to be 1.5 and 10^{-6} respectively.

4.2. Sampling Click Pairs

We generate user click pairs as pairwise ordinal constraints, using a sampling strategy similar to [37]. Specifically, we first divide the RGB input into regions via superpixel segmentation [1] and create a graph by connecting the centers of adjacent superpixels. We discard those connections whose edge lengths are either shorter than 5% or longer than 20% of the image diagonal length. For each superpixel region, we compute the ground truth median depth value and compare it with the median depth value predicted by the base network. The superpixel pairs whose relative median depth values are inconsistent between the ground truth and base network predictions are retained as click pair simulations.

Our sampling strategy preserves human perception, as our goal is to allow humans to interactively refine the depth predictions. In our implementation, sampled constraints are represented as $(0, +1, -1)$ ternary masks of input image size. We use $+1$ and -1 to indicate the ordinal relation between pixels in two superpixel regions, while 0 indicates no ordinal constraint exists on the pixel.

We apply the sampling strategy described above on both NYU Depth V2 and Stanford 2D-3D-S datasets. The simulated click pairs are sparse. On average, 5.8 click pairs are sampled on NYU Depth V2 and 12.6 click pairs are sampled on Stanford 2D-3D-S for each image. During training time, we randomly pick sampled click pairs as ordinal constraints into our system.

4.3. Evaluation

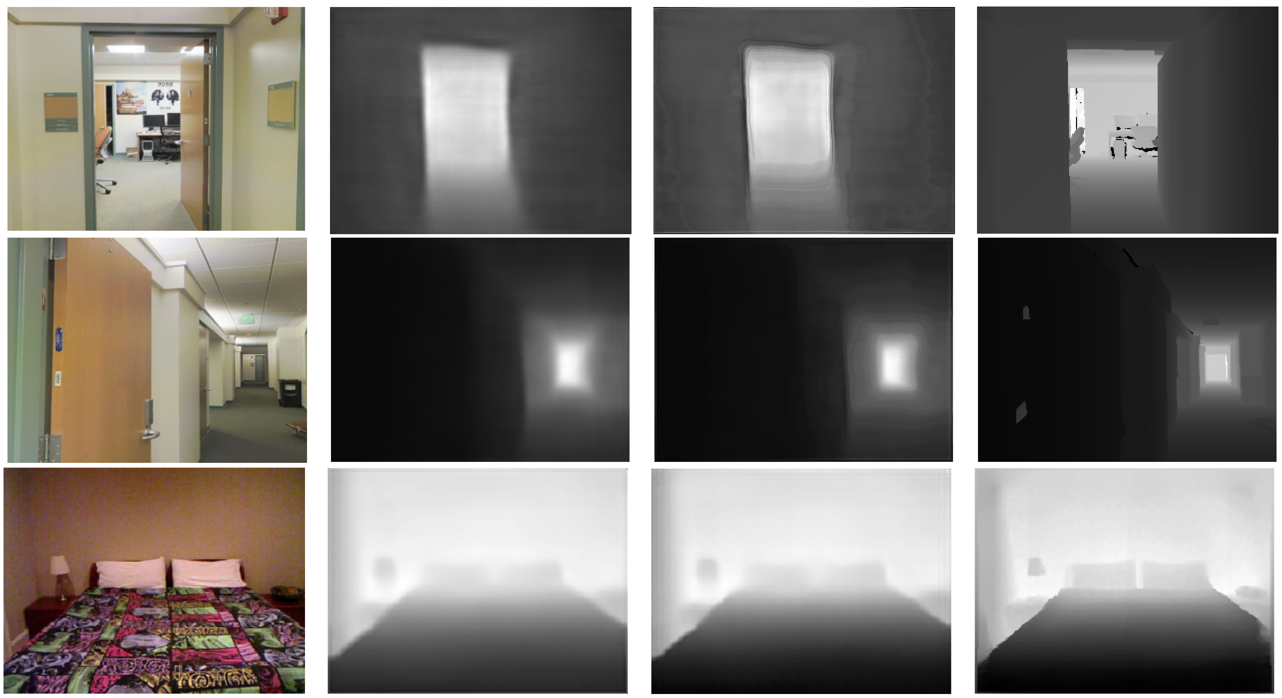
Evaluation criteria. We compute the results on both datasets using standard error metrics used by previous work, including *mean relative absolute error* (MRAE), *mean absolute error in log space* (Log10) and *root mean squared error* (RMSE). We show that our proposed method performs better quantitatively in terms of these error metrics. We also show that the depth estimation results have been improved qualitatively (Figure 5). In addition, we show that the performance of our system improves as more user click pairs are used.

We compare the performance of our proposed ADMM network against baseline results. **FCRN** are the results of the base network [15] evaluated on both datasets. **FCRN+ADMM(L1)** are the results of using ADMM modules with L1 smoothness priors on generated depth map. **FCRN+ADMM(L1+Ordinal)** are our results of using ADMM modules with one randomly sampled click pair on input image as ordinal constraints together with L1 smoothness priors. The quantitative results are summarized in Table 1.

Discussions. On both datasets, **FCRN+ADMM(L1)** improves over the baseline **FCRN** method. Compared with the output generated from base network, L1 terms help improve the quality at depth discontinuity boundaries and tend to make the output look more crisp (Figure 4). Among all the methods, **FCRN+ADMM(L1+Ordinal)** achieves the best results. We compare the visual quality of refined depths generated by **FCRN+ADMM(L1+Ordinal)** against

	NYU Depth V2			Stanford 2D-3D-S		
	MRAE	Log10	RMSE	MRAE	Log10	RMSE
Wang et al., [31]	0.220	0.094	0.745	–	–	–
Eigen and Fergus, [7]	0.158	–	0.641	–	–	–
Roy and Todorovic, [24]	0.187	0.078	0.744	–	–	–
FCRN, [15]	0.127	0.055	0.573	–	–	–
FCRN, [15]*	0.147	0.063	0.657	0.226	0.096	0.848
FCRN + ADMM (L1)	0.147	0.062	0.647	0.221	0.094	0.831
FCRN + ADMM (L1 + Ordinal)	**0.146**	**0.062**	**0.636**	**0.219**	**0.094**	**0.825**

Table 1: Depth estimation results on NYU Depth V2 (left) and Stanford 2D-3D-S (right) datasets. *Our own evaluation of the FCRN network on both of the datasets. We evaluated the TensorFlow model released by [15] on NYU Depth V2 dataset but were not able to reproduce the numbers reported in the paper.



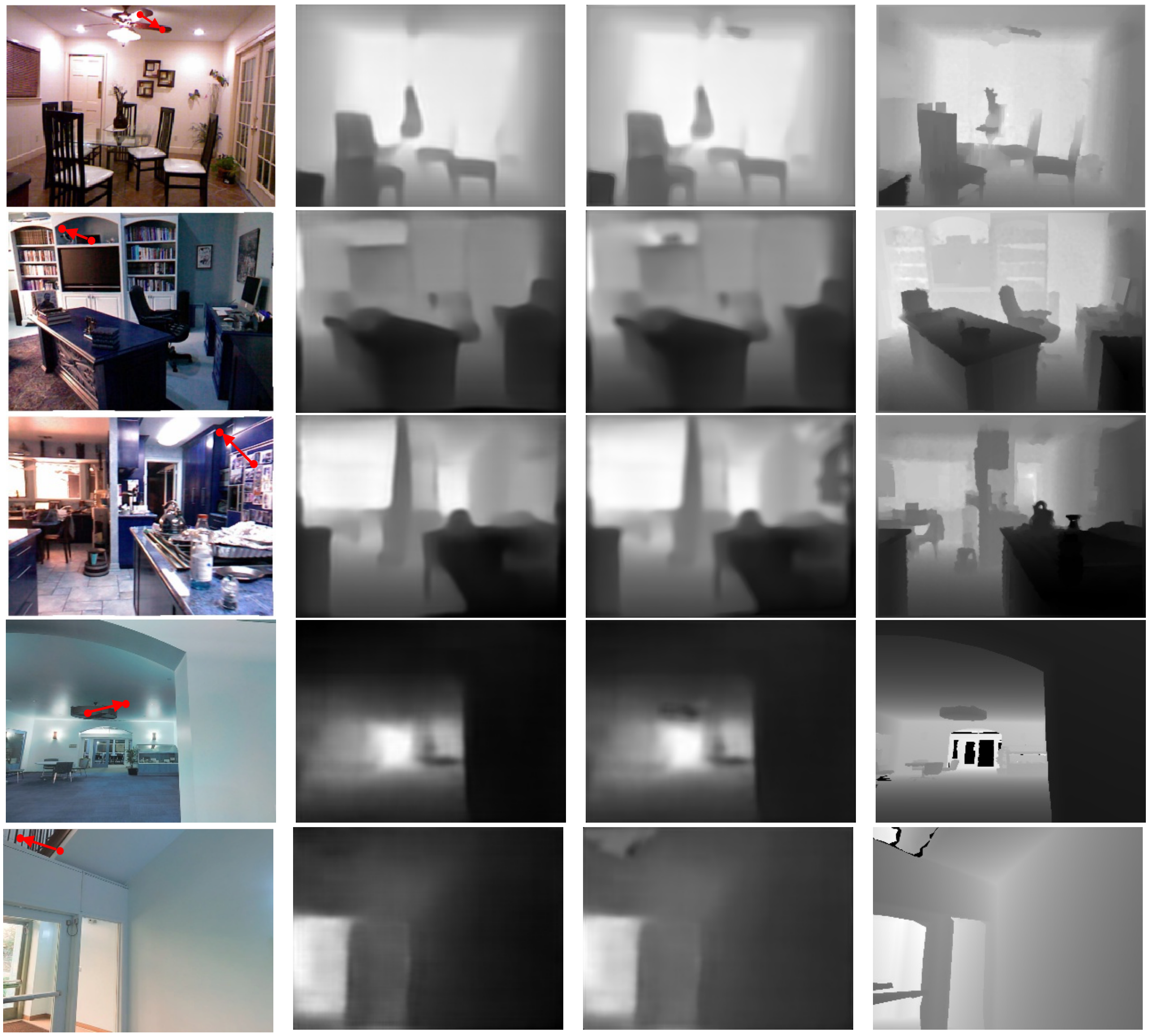
(a) Input (b) **FCRN** (c) **FCRN+ADMM(L1)** (d) Ground truth

Figure 4: Qualitative comparisons between (b) the base network output (**FCRN**) and (c) our ADMM network with L1 priors (**FCRN+ADMM(L1)**) on input images shown in (a).

the predicted depth generated by base network (Figure 5). We further experiment with different number of click pairs as extra input on NYU Depth V2 dataset (Figure 3). Our network structure is flexible enough to take any number of constraints as input at runtime. The performance of our system will be increasingly improved as more click pairs are added.

5. Conclusion

In this paper, we present an end-to-end neural network for monocular depth estimation that takes click pairs as extra input to allow for user interactions. We formulate such click pairs using pairwise ordinal constraints into a quadratic program problem and propose an ADMM solver to iteratively generate refined depth predictions constrained by such ordinal relations. Our proposed network shows competitive performance compared to state-of-the-art baselines on two challenging benchmark datasets. In future work, we will explore using prior terms with edge information or semantic segmentation masks from input images to integrate into our ADMM network modules. One limitation of our approach is that we need to pre-generate the click pairs for training. We will study how to use an active learning or human-in-the-loop approach to dynamically feed the network with ordinal constraints, that are generated on the fly during the training stage. Another interesting direction is to relax the hard constraints into weighted soft constraints, in case there are some inaccurate ordinal pairs from user interactions.



(a) Input (b) **FCRN** (c) Our results (d) Ground truth

Figure 5: Qualitative results on NYU Depth V2 (top three rows) and Stanford 2D-3D-S (bottom two rows) datasets. (a) Ordinal constraints are visualized as red arrows pointing from a closer click to another further point. We compare (c) our ADMM network with ordinal constraints and L1 priors (**FCRN+ADMM(L1+Ordinal)**) to (b) the **FCRN** base network.

Acknowledgement. We would like to thank Sam Hare for fruitful discussions, Aletta Hiemstra for proofreading our paper draft, and the anonymous reviewers for their valuable feedback and suggestions.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, et al. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. 5

[2] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 5

[3] S. Bell, K. Bala, and N. Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics*, 33(4):159:1–12, 2014. 1

[4] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems*, pages 730–738, 2016. 2, 3

[5] W. Chen, D. Xiang, and J. Deng. Surface normals in the wild. In *IEEE Conference on International Conference on Computer Vision*, pages 1557–1566, 2017. 2, 3

[6] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning

with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996. 2

[7] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE International Conference on Computer Vision*, pages 2650–2658, 2015. 1, 2, 3, 6

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 2

[9] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. 1

[10] H. Fu, M. Gong, C. Wang, and D. Tao. A compromise principle in deep monocular depth estimation. *arXiv preprint arXiv:1708.08267*, 2017. 1

[11] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 437–446, 2015. 2

[12] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013. 3

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[14] Y. Kuznietsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6647–6655, 2017. 1

[15] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *International Conference on 3D Vision*, pages 239–248, 2016. 1, 2, 3, 5, 6

[16] J. Li, R. Klein, and A. Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *IEEE Conference on International Conference on Computer Vision*, pages 22–29, 2017. 2, 3

[17] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2016. 3

[18] A. Mousavian, H. Pirsiavash, and J. Košecká. Joint semantic segmentation and depth estimation with deep convolutional networks. In *International Conference on 3D Vision*, pages 611–619, 2016. 2

[19] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760, 2012. 5

[20] S. Paris, P. Kornprobst, J. Tumblin, F. Durand, et al. Bilateral filtering: Theory and applications. *Foundations and Trends® in Computer Graphics and Vision*, 4(1):1–73, 2009. 3

[21] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018. 1, 2

[22] G. Riegler, D. Ferstl, M. Rüther, and H. Bischof. A deep primal-dual network for guided depth super-resolution. In *British Machine Vision Conference*, pages 7.1–7.14, 2016. 2

[23] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004. 1

[24] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5506–5514, 2016. 6

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[26] D. Singaraju and R. Vidal. Interactive image matting for multiple layers. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008. 1

[27] P. P. Srinivasan, R. Garg, N. Wadhwa, R. Ng, and J. T. Barron. Aperture supervision for monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6393–6401, 2018. 2

[28] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI Conference on Artificial Intelligence*, pages 4278–4284, 2017. 2

[29] J. T. Todd and J. F. Norman. The visual perception of 3-D shape from multiple cues: Are observers capable of perceiving metric structure? *Perception & Psychophysics*, 65(1):31–47, 2003. 2

[30] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning*, 2004. 2

[31] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2809, 2015. 2, 3, 6

[32] S. Wang, S. Fidler, and R. Urtasun. Proximal deep structured models. In *Advances in Neural Information Processing Systems*, pages 865–873, 2016. 2

[33] K. Xian, C. Shen, Z. Cao, H. Lu, Y. Xiao, R. Li, and Z. Luo. Monocular relative depth perception with web stereo data supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 311–320, 2018. 1

[34] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3917–3925, 2018. 1

[35] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–381, 2016. 1

[36] Y. Yang, J. Sun, H. Li, and Z. Xu. Deep ADMM-net for compressive sensing MRI. In *Advances in Neural Information Processing Systems*, pages 10–18, 2016. 2, 4

[37] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman. Learning ordinal relationships for mid-level vision. In *IEEE International Conference on Computer Vision*, pages 388–396, 2015. 2, 3, 5