Monocular Depth Estimation Using Relative Depth Maps

Jae-Han Lee
Korea University
jaehanlee@mcl.korea.ac.kr

Chang-Su Kim
Korea University
changsukim@korea.ac.kr

Abstract

We propose a novel algorithm for monocular depth estimation using relative depth maps. First, using a convolutional neural network, we estimate relative depths between pairs of regions, as well as ordinary depths, at various scales. Second, we restore relative depth maps from selectively estimated data based on the rank-1 property of pairwise comparison matrices. Third, we decompose ordinary and relative depth maps into components and recombine them optimally to reconstruct a final depth map. Experimental results show that the proposed algorithm provides the state-of-art depth estimation performance.

用卷积网络估计多尺度的区域间相对深度

基于秩1性质恢复相对深度

分解序和相对深度重组为深度

1. Introduction

Depth estimation is a fundamental problem of computer vision to estimate depth information of a scene from one or more images. Estimated depths give important geometric clues in vision applications, such as image synthesis [8,44], scene recognition [50, 56], pose estimation [60, 68], and robotics [4, 34]. There are various techniques for inferring depths from multi-view images [48, 55] or video sequences [30, 62], which provide promising results. However, when only a single image is available, the problem is challenging since it is ill-posed [12].

Early methods for monocular depth estimation made assumptions about scenes: a space composed of box blocks [16], a scene consisting of planar regions [54], a typical indoor room with a floor and walls [9, 32], and the dark channel prior [17]. However, these methods become unreliable when the assumptions are invalid.

In recent years, monocular depth estimation methods based on convolutional neural networks (CNNs) [6, 11–13, 31, 33, 51] have been proposed, with the advance in computing hardware and the availability of abundant training data [14, 57], improving the performance dramatically. Some methods [20, 36, 39, 65, 66] combine CNNs with conditional random field (CRF) models to yield more edge-conforming depth maps. Also, attempts have been made to estimate depths jointly with closely related data [27, 47, 61, 64, 69], such as surface normal and optical flow.

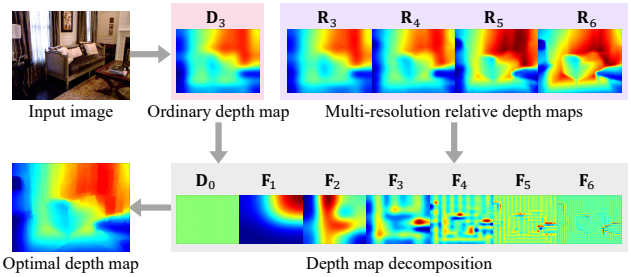

Figure 1. An overview of the proposed algorithm. First, one ordinary depth map and four relative depth maps are obtained from an image. Then, they are decomposed into depth components, which are, in turn, combined to reconstruct an optimal depth map.

These CNN-based methods attempt to estimate absolute depths directly. However, as noted in [12], monocular depth estimation is ambiguous in scale: an object may appear the same as another identically-shaped but smaller object in a nearer distance. On the other hand, the ratio between depths of two points, which is referred to as *relative depth* in this work, is scale-invariant. It is easier even for a human being to choose the nearer one between two points than to estimate the absolute depth of each point. In other words, relative depths are easier to estimate than ordinary depths.

Based on these observations, we propose a novel monocular depth estimation algorithm using relative depth maps. Figure 1 shows an overview of the proposed algorithm. First, we develop a CNN in the encoder-decoder architecture, which includes multiple decoder blocks for estimating relative depths, as well as ordinary depths, at various scales. Second, we form a pairwise comparison matrix, which is sparsely populated by the estimated relative depths. By exploiting the rank-1 property of the matrix, we restore the entire matrix using the alternating least squares (ALS) algorithm [28], from which a relative depth map is obtained. Third, each depth map is decomposed into components, which are re-combined to reconstruct a final depth map through a constrained optimization scheme. Experimental results show that the proposed algorithm provides the state-of-the-art depth estimation performance.

We highlight main contributions of this work as follows:

- We propose the notion of relative depth and develop an

efficient estimator for relative depth maps based on the rank-1 property of pairwise comparison matrices.

- We propose novel methods for depth map decomposition and depth component combination.
- We achieve the state-of-the art depth estimation performance on the NYUv2 dataset [57].

2. Related Work

Prior to the extensive adoption of CNNs, hand-crafted features were used for monocular depth estimation. Saxena *et al*. [53] proposed a Markov random field (MRF) model to estimate depths from multi-scale patches and global-scale column patches. Also, Saxena *et al*. [54] predicted depths by inferring plane parameters, assuming that a scene consists of planar regions. Liu *et al*. [38] exploited the *a priori* knowledge of semantic segmentation classes to predict depths. Karsch *et al*. [25] assumed that semantically similar images have similar depth distributions. They estimated depth maps by searching similar images from a database and warping them.

Recently, various CNN-based techniques for monocular depth estimation have been proposed. Eigen *et al*. [12] used the AlexNet structure [29] for global depth prediction and an additional fine scale network for local depth refinement. Eigen and Fergus [11] extended the method in [12] to three levels and performed depth estimation, normal estimation, and semantic segmentation jointly. Roy and Todorovic [51] proposed a depth estimation model to incorporate shallow CNNs into a regression forest. Laina *et al*. [31] developed a depth estimation network based on the ResNet structure [19] and also proposed an up-projection module to increase depth map resolutions. Fu *et al*. [13] proposed the deep ordinal regression network (DORN), which transforms the depth regression task into a classification problem. Their algorithm yielded the state-of-the-art depth estimation performance.

To generate sharper and more edge-conforming depth maps, conditional random field (CRF) models are often combined with CNNs. Li *et al*. [36] estimated depth information at the superpixel level using a CNN and refined it at the pixel level based on a CRF model. Liu *et al*. [40] developed another superpixel-based algorithm. They trained unary and pairwise terms of CRF within a CNN framework. Xu *et al*. [65] extracted feature maps at several CNN layers, performed CRF optimization at those layers to yield multiple depth maps, and integrated them into a final depth map. Heo *et al*. [20] predicted depths and also the corresponding reliability levels. They exploited the reliability information in the CRF optimization. Xu *et al*. [66] integrated multi-scale CRF optimization into an encoder-decoder network, enabling end-to-end training.

Extending the domain of training data tends to have positive impacts on the estimation performance of a deep network. Therefore, some methods utilize additional annotation data to train depth estimation networks. For instance, Wang *et al*. [61] proposed a joint CNN structure for depth map estimation and semantic segmentation. Moreover, they improved depth estimation results via CRF optimization. Qi *et al*. [47] utilized the geometric relationship between surface normals and depths, improving the results of both normal and depth estimation. Also, Yin and Shi [69] proposed a joint estimation algorithm for depths, optical flow, and camera motion.

The methods in [7, 70] are similar to the proposed algorithm in that they also use pairwise depth comparison results between pixels for monocular depth estimation. Zoran *et al*. [70] predicted relative depths between sampled points and propagated them to superpixels to reconstruct an entire depth map. Chen *et al*. [7] categorized relative depths between pixels into three classes: "closer," "further," and "equal." They obtained pixel-level predictions by training their network with different loss functions according to pairwise labels. The proposed algorithm, however, is different from [7, 70]. While [7, 70] use comparison results between coarsely sampled points, the proposed algorithm estimates dense pairwise information and combines it with ordinary depth maps to reconstruct fine scale depth information.

3. Proposed Algorithm

3.1. Depth Map Decomposition

Let $\mathbf{I} \in \mathbb{R}^{r \times c}$ be an image of size $r \times c$. The goal is to estimate the corresponding depth map $\mathbf{D} \in \mathbb{R}^{r \times c}$. However, this monocular depth estimation is ill-posed. Especially, it is ambiguous in scale [12]. For instance, a building and its small replica may produce an identical image, but have different depth maps. Even though we can predict the scale of an image approximately by learning from many training images, the ambiguity still remains. To address this issue, in this work, we define and estimate a scale-invariant quantity, called *relative depth*, which is the ratio between the depths of two regions in an image.

深度估计是病态的，一个建筑和他的缩小版本有相同的图像，但是不同的深度。尽管深度学习可以逼近训练集，但是这种歧义仍然存在。因此我们采用两个区域深度的比例=相对深度

If we know the relative depths of all pixel pairs in an image, we can reconstruct the depth map with a normalized scale. Before proving this, let us denote the geometric mean of a depth map $\mathbf{D}$ by

几何均值

$$g(\mathbf{D}) = \prod_{i=1}^{r} \prod_{j=1}^{c} \mathbf{D}(i,j)^{\frac{1}{rc}} \tag{1}$$

where $\mathbf{D}(i,j)$ is the (i,j)th depth in $\mathbf{D}$.

Proposition 1. *If the relative depths of all pixel pairs in* $\mathbf{I}$ *is known, then a scaled depth map* $\mathbf{D}/g(\mathbf{D})$ *can be reconstructed.* 所有点对的相对深度都有，那么就可以用D/几何均值重构

Proof. By assumption, for any pixel (i, j), we know all relative depths in $\mathbf{D}/\mathbf{D}(i, j)$. By averaging these depths geometrically, we obtain $g(\mathbf{D})/\mathbf{D}(i, j)$. Therefore, we know its reciprocal $\mathbf{D}(i, j)/g(\mathbf{D})$. Then, we have $\mathbf{D}/g(\mathbf{D})$. □

倒数

In fact, $\mathbf{D}/g(\mathbf{D})$ has the normalized scale as follows.

Proposition 2. *The geometric mean of* $\mathbf{D}/g(\mathbf{D})$ *is 1.*

Proof. $g(\mathbf{D}/g(\mathbf{D})) = g(\mathbf{D})/g(\mathbf{D}) = 1$. □

According to **Propositions 1** and **2**, if we know all relative depths between pixel pairs, we can reconstruct the relative depth map

只不过取归一化系数是几何均值

$$\mathbf{R} = \mathbf{D}/g(\mathbf{D}), \tag{2}$$

which is referred to as the *relative depth map*. Then, the relationship between the original depth map $\mathbf{D}$ and the relative depth map $\mathbf{R}$ can be rewritten as $\mathbf{D} = g(\mathbf{D})\mathbf{R}$.

Next, we reduce the depth map $\mathbf{D}$ to several sizes. Let $\mathbf{D}_n$ denote the depth map of size $2^n \times 2^n$. A lower resolution depth map $\mathbf{D}_{n-1}$ is obtained from $\mathbf{D}_n$ via

即四个格子深度的几何均值

$$\mathbf{D}_{n-1}(i, j) = \prod_{k=0}^{1}\prod_{l=0}^{1}\mathbf{D}_n(2i-k, 2j-l)^{\frac{1}{4}}. \tag{3}$$

In other words, a depth in $\mathbf{D}_{n-1}$ is the geometric mean of the four corresponding depths in $\mathbf{D}_n$. Note that the lowest resolution map $\mathbf{D}_0$ consists of a single depth, which equals the overall geometric mean $g(\mathbf{D})$.

D0=g(D)

In a typical depth map, low frequency components are more dominant [33]. Thus, their estimation affects depth reconstruction more strongly than the estimation of high frequency components. We regard $\mathbf{D}_{n-1}$ as low frequency information, which is obtained by eliminating high frequency (or fine detail) information in $\mathbf{D}_n$. Let $\mathbf{F}_n$ denote the fine detail map. First, we define the upsampling operation U to double the size of a depth map horizontally and vertically. It repeats each input depth four times to fill in the corresponding four pixels in the output depth map. Then, $\mathbf{F}_n$ is given by

$$\mathbf{F}_n = \mathbf{D}_n \oslash U(\mathbf{D}_{n-1}) \tag{4}$$

where $\oslash$ denotes the Hadamard division, *i.e.* element-wise division, of two matrices. Equivalently,

$$\mathbf{D}_n = U(\mathbf{D}_{n-1}) \otimes \mathbf{F}_n \tag{5}$$

where $\otimes$ is the Hadamard product.

Proposition 3. $\prod_{k=0}^{1}\prod_{l=0}^{1}\mathbf{F}_n(2i-k, 2j-l)^{\frac{1}{4}} = 1$ *for each* (i, j)*, and* $g(\mathbf{F}_n) = 1$.

Proof. It comes from (3) and (4). □

Table 1. Decomposition results of depths maps $\mathbf{D}_n$ and $\mathbf{R}_n$ for $3 \le n \le 7$.

	$\mathbf{D}_0$	$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$	$\mathbf{F}_4$	$\mathbf{F}_5$	$\mathbf{F}_6$	$\mathbf{F}_7$
$\mathbf{D}_3$	√	√	√	√	-	-	-	-
$\mathbf{D}_4$	√	√	√	√	√	-	-	-
$\mathbf{D}_5$	√	√	√	√	√	√	-	-
$\mathbf{D}_6$	√	√	√	√	√	√	√	-
$\mathbf{D}_7$	√	√	√	√	√	√	√	√
$\mathbf{R}_3$	-	√	√	√	-	-	-	-
$\mathbf{R}_4$	-	√	√	√	√	-	-	-
$\mathbf{R}_5$	-	√	√	√	√	√	-	-
$\mathbf{R}_6$	-	√	√	√	√	√	√	-
$\mathbf{R}_7$	-	√	√	√	√	√	√	√

In a logarithmic scale, $\mathbf{D}_n$ can be decomposed through the recursive application of (5),

$$\log \mathbf{D}_n = \log U^n(\mathbf{D}_0) + \sum_{i=1}^{n}\log U^{n-i}(\mathbf{F}_i) \tag{6}$$

where log is an element-wise logarithmic function. In other words, $\log \mathbf{D}_n$ is decomposed into the mean depth map $\log U^n(\mathbf{D}_0)$ and the residual depth maps $\log U^{n-i}(\mathbf{F}_i)$ for $1 \le i \le n$. Note that, by **Proposition 3**, the arithmetic mean of each residual map $\log U^{n-i}(\mathbf{F}_i)$ is zero. Similarly, the relative depth map $\mathbf{R}_n$ can be decomposed as

$$\log \mathbf{R}_n = \sum_{i=1}^{n}\log U^{n-i}(\mathbf{F}_i). \tag{7}$$

In this work, given an image $\mathbf{I}$, we estimate $\mathbf{D}_n$ and $\mathbf{R}_n$ for $3 \le n \le 7$. Then, we decompose each $\mathbf{D}_n$ or $\mathbf{R}_n$ via (6) or (7), respectively. Table 1 lists the decomposition results of these depth maps. Note that each component has multiple candidates. For example, $\mathbf{F}_1$ has 10 candidates in total, while $\mathbf{F}_6$ has 4. We combine the candidates to yield the optimal depth component, as described in Section 3.4 and in a supplemental document. Finally, we use the optimal components to generate the optimal depth map $\mathbf{D}_7$ via (6).

3.2. Depth Estimation Network

We use the encoder-decoder architecture [2, 67] to estimate depth maps, as shown in Figure 2. In the encoder part, deep features are extracted from an image. In the decoder part, up to ten decoders use these features to reconstruct ordinary depth maps $\mathbf{D}_n$ and relative depth maps $\mathbf{R}_n$.

Encoder part: The encoder processes an image to yield low-resolution, high-level features. DenseNet-BC [23], excluding the last dense block, is used as the encoder, which consists of one convolution layer, one max pooling layer, and three pairs of dense block and transition layer, as shown in Figure 2. Note that the last dense block in DenseNet-BC is employed in the ten decoders in the decoder part.

Each dense block in DenseNet-BC is defined by hyper-parameters: the number n of composite functions and the growth rate k. The settings of the dense blocks, including the hyper-parameters, are described in the supplemental

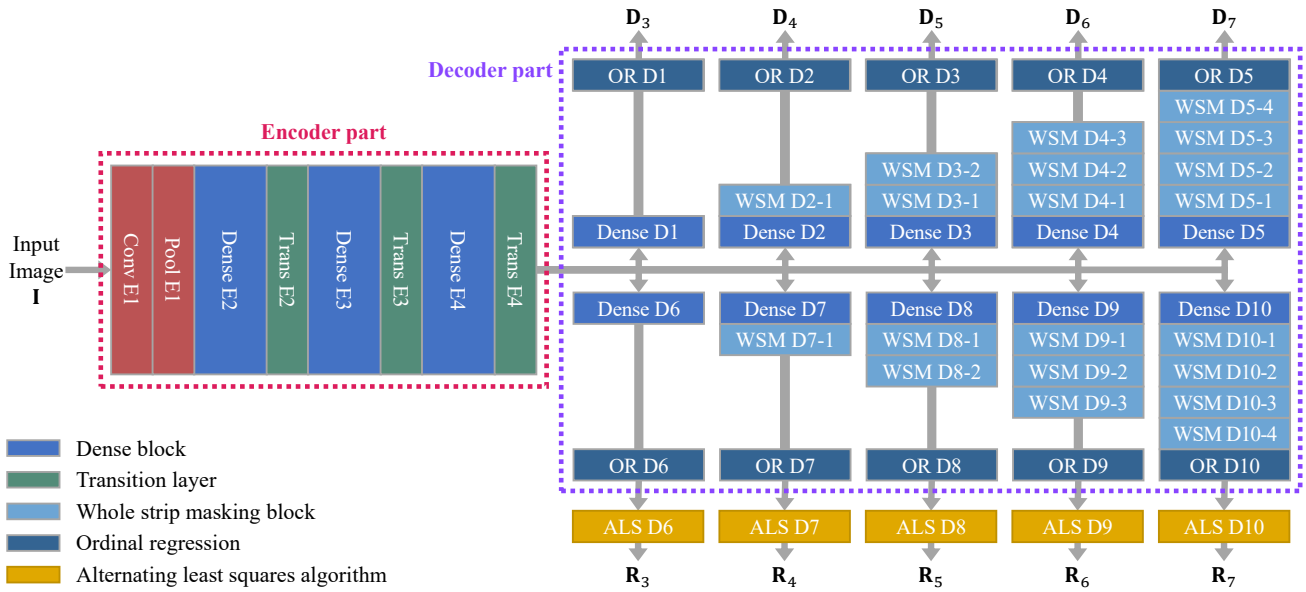

Figure 2. The structure of the proposed depth estimation network. As shown above, up to ten decoders can be used. In the default setting, the five decoders for ($\mathbf{D}_3$, $\mathbf{R}_3$, $\mathbf{R}_4$, $\mathbf{R}_5$, $\mathbf{R}_6$) are employed. WSM represents a whole strip masking block [20], OR an ordinal regression layer, and ALS an alternating least squares layer.

document. Overall, given an 224×224 RGB image, the encoder generates an 8×8 feature map with 1,056 channels.

Decoder part: The ten decoders are used to expand the low-resolution features to higher-resolution depth maps $\mathbf{D}_n$ and $\mathbf{R}_n$. Each decoder has one dense block and a variable number (0 to 4) of whole strip masking (WSM) blocks [20].

WSM is an up-sampling block in the inception structure [58, 59]. It increases the receptive field greatly, by applying kernels whose horizontal or vertical sizes equal those of an entire input signal. It has five inception paths, which use convolution kernels of sizes 1×1, 3×3, 5×5, $W \times 3$, and $3 \times H$, respectively. Here, W and H denote the width and height of an input signal.

The resolution of a target depth map determines the number of WSM blocks. For example, the decoders for estimating $\mathbf{D}_3$ and $\mathbf{R}_3$ include no WSM block, since $\mathbf{D}_3$ and $\mathbf{R}_3$ have 8×8 resolution that is equal to the resolution of the encoder feature map. On the other hand, the decoders for $\mathbf{D}_7$ and $\mathbf{R}_7$ use 4 WSM blocks, respectively, to extend the feature map to 128×128 resolution.

Ordinal regression: Each decoder performs ordinal regression [37] to reconstruct depths. An ordinal regression task can be carried out using multiple binary classifiers, which determine if a value is greater than different thresholds, respectively. Various ordinal regression methods have been proposed to solve regression problems [13, 24, 46]. In particular, Fu *et al*. [13] proposed a regression network, called DORN, for monocular depth estimation. For ordinal regression, they quantized a depth into a number of reconstruction levels using the space-increasing discretization scheme. We adopt their reconstruction levels and ordinal loss function in the decoders for ordinary depth maps $\mathbf{D}_n$.

However, in the decoders for relative depth maps $\mathbf{R}_n$, it is necessary to use a different set of reconstruction levels. Note that a relative depth is a ratio of two depths. Thus, for any relative depth r, there is always a reciprocal one $1/r$. In other words, in a logarithmic scale, the distribution of relative depths is symmetric with respect to zero. To determine reconstruction levels for $\mathbf{R}_3$, we compute the depth ratios for all pixel pairs from training data. We apply the Lloyd algorithm [42] to quantize them. To exploit the symmetry, we perform the algorithm only for the ratios greater than or equal to 1. Then, we fix 1 as one reconstruction level to conform to the symmetry, and determine 20 more reconstruction levels by alternating the nearest neighbor partitioning and the centroid computation [15]. Their reciprocals also become reconstruction levels. In total, there are 41 reconstruction levels. Also, reconstruction levels for $\mathbf{R}_n$ for $4 \leq n \leq 7$ are set to half the level interval of $\mathbf{R}_{n-1}$.

Relative depths can be estimated for all pairs of pixels. This, however, demands excessive complexity, since $(2^n \times 2^n) \times (2^n \times 2^n) = 2^{4n}$ pairs should be considered in $\mathbf{D}_n$. To reduce the complexity, for each pixel in $\mathbf{D}_n$, we estimate the depth ratios with respect to the neighboring 3×3 pixels only, reducing the number of pairs to $3^2 \times 2^{2n}$. Furthermore, these neighboring 3×3 pixels are selected from $\mathbf{D}_{n-1}$, instead of $\mathbf{D}_n$, as shown in Figure 3. This is advantageous, since each depth in $\mathbf{D}_n$ is compared with a larger region for a fixed number of comparisons. Unestimated relative depths are reconstructed using the ALS algorithm, as detailed in Section 3.3.

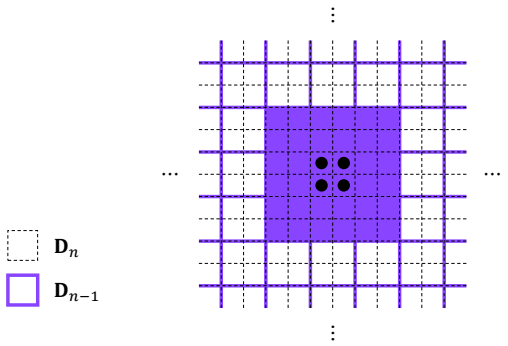

Figure 3. To estimate relative depths, each depth in $\mathbf{D}_n$, depicted by a dot, is compared with the depths of the 3×3 nearest pixels in $\mathbf{D}_{n-1}$, which are depicted by purple squares. For the illustration, $\mathbf{D}_n$ is overlaid with $\mathbf{D}_{n-1}$.

3.3. Relative Depth Map Reconstruction

In Figure 2, the bottom decoders $6 \sim 10$ estimate relative depth maps $\mathbf{R}_n$, $3 \leq n \leq 7$, respectively. To reduce the complexity, they estimate relative depths selectively. The remaining relative depths are reconstructed as follows.

First, in decoder 6, the relative depths for all pixel pairs in the lowest-resolution depth map $\mathbf{D}_3$ are estimated. Inevitably, there are estimation errors. Let us consider three pixels $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$. The decoder estimates relative depths $\frac{\mathbf{D}_3(\mathbf{i})}{\mathbf{D}_3(\mathbf{j})}$, $\frac{\mathbf{D}_3(\mathbf{j})}{\mathbf{D}_3(\mathbf{k})}$, and $\frac{\mathbf{D}_3(\mathbf{i})}{\mathbf{D}_3(\mathbf{k})}$. However, due to estimation errors, the results may be inconsistent, *i.e.* it is possible that $\frac{\mathbf{D}_3(\mathbf{i})}{\mathbf{D}_3(\mathbf{j})} \times \frac{\mathbf{D}_3(\mathbf{j})}{\mathbf{D}_3(\mathbf{k})} \neq \frac{\mathbf{D}_3(\mathbf{i})}{\mathbf{D}_3(\mathbf{k})}$. We should process the estimated relative depth to yield consistent and reliable results.

To this end, we construct the pairwise comparison matrix $\mathbf{P}_3$, which contains the relative depths between all pixel pairs in $\mathbf{D}_3$. Since the number of pixels in $\mathbf{D}_3$ is 8×8, the size of $\mathbf{P}_3$ is 64×64. The (i, j)th element $\mathbf{P}_3$ is given by the estimate of d_j/d_i, where d_i denotes the ith depth in the reshaped vector of $\mathbf{D}_3$.

Proposition 4. *If there is no estimation error, $\mathbf{P}_3$ is a rank-1 matrix.*

Proof. In the ideal case with no error, we have $\mathbf{P}_3 = [d_1, d_2, \cdots, d_{64}]^T[\frac{1}{d_1}, \frac{1}{d_2}, \cdots, \frac{1}{d_{64}}]$. □

If there are errors, Saaty [52] showed that the principal eigenvector corresponding to the largest eigenvalue of $\mathbf{P}_3$ is a good approximation of $[d_1, d_2, \cdots, d_{64}]^T$ up to a scale factor. Note that, by the Perron-Frobenius theorem [21], since $\mathbf{P}_3$ is positive, the largest eigenvector is algebraically simple and positive and all elements in the principal eigenvector are also positive. Thus, by normalizing the principal eigenvector so that the geometric mean of elements is 1, we reconstruct the relative depth map $\mathbf{R}_3$.

To reconstruct $\mathbf{R}_n$ for $4 \leq n \leq 7$, we should redefine the comparison matrix, since depths in $\mathbf{D}_n$ are compared with those in $\mathbf{D}_{n-1}$ as shown in Figure 3. Similar to $\mathbf{P}_3$ in the proof of **Proposition 4**, in the ideal case, the comparison matrix is given by

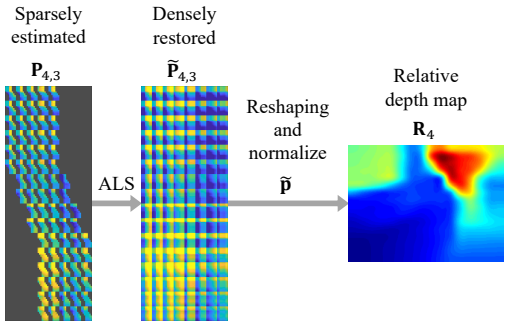

Figure 4. A sparse comparison matrix $\mathbf{P}_{4,3}$ is restored to a dense matrix $\tilde{\mathbf{P}}_{4,3}$ by the ALS algorithm. Then, $\tilde{\mathbf{P}}_{4,3}$ is reshaped and normalized to a relative depth map $\mathbf{R}_4$.

$$\mathbf{P}_{n,n-1} = [d_1^n, d_2^n, \cdots, d_{2^{2n}}^n]^T[\frac{1}{d_1^{n-1}}, \frac{1}{d_2^{n-1}}, \cdots, \frac{1}{d_{2^{2n-2}}^{n-1}}] \tag{8}$$

where d_i^n denotes the ith depth in the reshaped vector of $\mathbf{D}_n$. Without estimation errors, the rank of $\mathbf{P}_{n,n-1}$ is also 1. When there are estimation errors, the eigenvalue decomposition method for reconstructing $\mathbf{R}_3$ cannot be used in this case because $\mathbf{P}_{n,n-1}$ is not a square matrix. Instead, we may use singular value decomposition (SVD). It is known that

$$\hat{\mathbf{P}}_{n,n-1} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T \tag{9}$$

is the best rank-1 approximation of $\mathbf{P}_{n,n-1}$ [5], where σ_1 is the largest singular value, and $\mathbf{u}_1$ and $\mathbf{v}_1$ are the corresponding singular vectors. Therefore, $\mathbf{R}_n$ can be obtained by normalizing the left singular vector $\mathbf{u}_1$.

However, as shown in Figure 3, only a portion of relative depths, d_i^n/d_j^{n-1}, are estimated and $\mathbf{P}_{n,n-1}$ is incomplete. The missing entries of $\mathbf{P}_{n,n-1}$ should be filled in appropriately before the rank-1 approximation. Various algorithms [26, 49] have been proposed to solve this matrix completion problem. We employ the ALS algorithm [28] as follows. Let $\mathcal{S}$ denote the set of positions (r, c) in $\mathbf{P}_{n,n-1}$, where the relative depths are estimated by the decoder. Also, let $\mathbf{p}$ and $\mathbf{q}$ be vectors of size 2^{2n} and 2^{2n-2}, respectively. Then, we repeat the following two steps alternately.

$$\mathbf{q} \leftarrow \arg\min_{\mathbf{q}} \sum_{(r,c)\in\mathcal{S}} \left(\mathbf{p}(r)\mathbf{q}(c) - \mathbf{P}_{n,n-1}(r,c)\right)^2 \tag{10}$$

$$\mathbf{p} \leftarrow \arg\min_{\mathbf{p}} \sum_{(r,c)\in\mathcal{S}} \left(\mathbf{p}(r)\mathbf{q}(c) - \mathbf{P}_{n,n-1}(r,c)\right)^2 \tag{11}$$

In each step, the convex condition is satisfied and the closed form solution for $\mathbf{q}$ or $\mathbf{p}$ is easily derived. Thus, the algorithm yields convergent solutions $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{q}}$, and the approximation

$$\tilde{\mathbf{P}}_{n,n-1} = \tilde{\mathbf{p}}\tilde{\mathbf{q}}^T \tag{12}$$

is obtained. Notice that this is already a rank-1 approximation of $\mathbf{P}_{n,n-1}$. Therefore, we reconstruct the relative depth map $\mathbf{R}_n$ by normalizing and reshaping the left vector $\tilde{\mathbf{p}}$. Figure 4 shows this process of filling a sparse $\mathbf{P}_{n,n-1}$ and restoring a relative depth map $\mathbf{R}_n$.

3.4. Depth Component Combination

In general, an ordinary depth map reconstructs the overall depth distribution robustly, while relative depth maps are better for estimating fine details. Also, depending on the resolution, each relative depth map estimates depth information at a certain scale reliably. Thus, by combining all these maps at multiple resolutions, we obtain a faithful depth map that takes advantages of those component maps.

We estimate up to ten depth maps, $\mathbf{D}_n$ and $\mathbf{R}_n$ for $3 \leq n \leq 7$, each of which is decomposed into components, as listed in Table 1. Since there are multiple candidates for each component, we obtain an optimal estimate by linearly combining them in a logarithmic domain. For the optimal combination, we minimize the mean squared error, subject to constraints on weighting parameters (*e.g.* nonnegativity of weights), using the interior point method [1]. Then, we use these optimal components to generate a final depth map via (6). The optimal combination method is described in more detail in the supplemental document.

4. Experimental Results

4.1. Dataset and Evaluation Protocol

We assess the performance of the proposed algorithm on the NYUv2 dataset [57]. It includes indoor video sequences, composed of RGB images of spatial resolution 480×640 and the corresponding depth maps captured with Microsoft Kinect devices. A captured depth map has missing regions, and the method in [35] is used to fill in those regions. We use all training sequences to train the proposed algorithm and employ the 654 test RGBD images for evaluation. Also, we valid-crop the test images to spatial resolution 427×561, as done in [6, 33, 41]. For quantitative assessment of depth maps, we use seven metrics in Table 2 [10, 12, 31]. Among them, the Spearman's ρ is the correlation coefficient between the ranks of estimated depths and ground-truth depths [10]. It measures how well an estimated depth map preserves the ordering (or ranks) of pixel depths in the ground-truth depth map.

KITTI [14] is another dataset widely used for evaluating monocular depth estimation algorithms. We show that the proposed algorithm provides competitive performances also on KITTI in the supplemental document.

4.2. Network Training

We initialize the network parameters as done in [18] and optimize them using the Nestrov method [45]. We set the initial learning rate, momentum, and weight decay to 10^{-5}, 0.9, and 10^{-4}, respectively. Also, we adjust the learning rate based on the repetitive shifted cosine function [22, 43]. We set the cycle of the cosine function to $1/4$ epoch.

We train the network in two steps. First, we train the encoder with a single decoder, which is the decoder for generating $\mathbf{D}_3$ in Figure 2. Second, after fixing the encoder parameters, we train each of the ten decoders. We set the batch size to 4, except for the decoder for $\mathbf{R}_7$, for which the batch size is 2 due to the limited GPU memory.

Table 2. Evaluation metrics for estimated depth maps: $\hat{d}_i$ and d_i denote estimated and ground-truth depths of pixel i, respectively, and N is the number of pixels in a depth map.

Metric	Definition
RMSE (lin)	$(\frac{1}{N}\Sigma_i(\hat{d_i} - d_i)^2)^{\frac{1}{2}}$
RMSE (log)	$(\frac{1}{N}\Sigma_i(\log \hat{d_i} - \log d_i)^2)^{\frac{1}{2}}$
RMSE (s.inv)	RMSE (log) for relative depth maps
ARD	$\frac{1}{N}\Sigma_i \lvert\hat{d_i} - d_i\rvert / d_i$
SRD	$\frac{1}{N}\Sigma_i \lvert\hat{d_i} - d_i\rvert^2 / d_i$
$\delta < t$	Percentage of d_i such that $\max\{\frac{\hat{d_i}}{d_i}, \frac{d_i}{\hat{d_i}}\} < t$
Spearman's ρ	Correlation coefficient $\in [-1, 1]$ between the ranks of $\{\hat{d_i}\}$ and $\{d_i\}$

4.3. Comparison with the State-of-the-Arts

Table 3 compares the proposed algorithm with conventional algorithms [3, 6, 7, 11–13, 31, 33, 36, 39, 41, 63, 66, 70] on the NYUv2 dataset. Some algorithms use different methods for depth map cropping and performance measurement. Therefore, for a fair comparison, we adopted the evaluation scheme of [6, 33, 41] as the common method and attempted to follow it as closely as possible. Specifically, for the algorithms in [6, 11–13, 31, 33, 39], the result depth maps, provided by the respective authors or generated by the source codes by the authors, are evaluated by the common method. For the other algorithms, the performance scores are excerpted directly from the papers.

It can be observed from Table 3 that, in terms of 6 (out of 8) metrics, the proposed algorithm outperforms all the conventional algorithms using only the NYUv2 RGBD training data. In the other two metrics, ARD and ($\delta < 1.25$), the proposed algorithm yields the third best and the second best performances, respectively. Especially, the proposed algorithm provides a significantly higher ρ than the conventional algorithms. This means that the proposed algorithm predicts the depth orders of pixels more accurately by estimating relative depth maps, containing order information, as well as ordinary depth maps.

Figure 5 shows qualitative comparison results. As compared with the conventional algorithms [13, 31], the proposed algorithm provides more accurate depth maps with less errors. Even though Fu *et al*. [13] yield smaller errors than Laina *et al*. [31], their errors have disorderly patterns and thus their depth maps look noisier. In contrast,

Table 3. Performance comparison on the NYUv2 test data. The best results are boldfaced, and the second best ones are underlined. Note that we reevaluate some algorithms [6, 11–13, 31, 33, 39] by using the evaluation scheme of [6, 33, 41].

	The lower, the better					The higher, the better			
	RMSE (lin)	RMSE (log)	RMSE (s.inv)	ARD	SRD	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	ρ
Zoran *et al*. [70]	1.200	0.420	-	0.400	0.540	-	-	-	-
Chen *et al*. [7]	1.110	0.380	0.450	0.350	0.430	-	-	-	-
Liu *et al*. [41]	1.080	-	-	0.327	-	-	-	-	-
Baig *et al*. [3]	0.802	-	-	0.241	-	61.0%	-	-	-
Li *et al*. [36]	0.821	-	-	0.232	-	62.1%	88.6%	96.8%	-
Eigen *et al*. [12]	0.874	0.284	0.219	0.218	0.207	61.6%	88.9%	97.1%	0.800
Liu *et al*. [39]	0.756	0.261	0.214	0.209	0.180	66.2%	91.3%	97.9%	0.786
Eigen and Fergus [11]	0.639	0.215	0.171	0.158	0.121	77.1%	95.0%	98.8%	0.886
Xian *et al*. [63]	0.660	-	-	0.155	-	78.1%	95.0%	98.7%	-
Xu *et al*. [66]	0.593	-	-	0.125	-	80.6%	95.2%	98.6%	-
Chakrabarti *et al*. [6]	0.620	0.205	0.166	0.149	0.118	80.6%	95.8%	98.7%	0.902
Laina *et al*. [31]	0.584	0.198	0.164	0.136	0.101	82.2%	95.6%	98.9%	0.887
Lee *et al*. [33]	0.572	0.193	0.156	0.139	0.096	81.5%	96.3%	99.1%	0.899
Fu *et al*. [13]	0.547	0.188	0.158	**0.116**	0.089	**85.6%**	96.1%	98.6%	0.899
Proposed	**0.538**	**0.180**	**0.148**	0.131	**0.087**	83.7%	**97.1%**	**99.4%**	**0.914**

Table 4. Ablation study using various combinations of depth maps. We use five maps ($\mathbf{D}_3$, $\mathbf{R}_3$, $\mathbf{R}_4$, $\mathbf{R}_5$, $\mathbf{R}_6$) in the default mode.

Used ordinary depth map					Used relative depth map					The lower, the better		The higher, the better			
$\mathbf{D}_3$	$\mathbf{D}_4$	$\mathbf{D}_5$	$\mathbf{D}_6$	$\mathbf{D}_7$	$\mathbf{R}_3$	$\mathbf{R}_4$	$\mathbf{R}_5$	$\mathbf{R}_6$	$\mathbf{R}_7$	RMSE (lin)	ARD	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	ρ
√	-	-	-	-	-	-	-	-	-	0.583	0.143	81.2%	96.3%	99.2%	0.885
-	√	-	-	-	-	-	-	-	-	0.556	0.135	82.8%	96.9%	99.3%	0.901
-	-	√	-	-	-	-	-	-	-	0.553	0.134	83.1%	96.9%	99.3%	0.903
-	-	-	√	-	-	-	-	-	-	0.552	0.133	83.1%	96.9%	99.3%	0.904
-	-	-	-	√	-	-	-	-	-	0.552	0.133	83.1%	97.0%	99.3%	0.904
√	√	-	-	-	-	-	-	-	-	0.555	0.135	82.8%	96.9%	99.3%	0.901
√	√	√	-	-	-	-	-	-	-	0.551	0.134	83.1%	96.9%	99.3%	0.903
√	√	√	√	-	-	-	-	-	-	0.550	0.133	83.2%	97.0%	99.3%	0.904
√	√	√	√	√	-	-	-	-	-	0.550	0.133	83.0%	97.0%	99.3%	0.905
√	-	-	-	-	√	-	-	-	-	0.580	0.142	81.3%	96.4%	99.2%	0.889
√	-	-	-	-	√	√	-	-	-	0.549	0.134	83.1%	97.0%	99.4%	0.907
√	-	-	-	-	√	√	√	-	-	0.540	0.132	83.6%	97.1%	99.4%	0.912
√	-	-	-	-	√	√	√	√	-	0.538	0.131	83.7%	97.1%	99.4%	0.914
√	-	-	-	-	√	√	√	√	√	0.538	0.131	83.7%	97.2%	99.4%	0.914
√	√	√	√	√	√	√	√	√	√	0.539	0.130	83.8%	97.1%	99.4%	0.912

the proposed algorithm provides cleaner depth maps and outperforms the conventional algorithms both quantitatively and qualitatively. Also, figure 6 compares the 3D visualization results of depth maps. Again, the proposed algorithm shows more reliable results than the conventional algorithms [13, 31].

4.4. Ablation Study

The proposed algorithm uses up to ten decoders in Figure 2 to generate ordinary depth maps $\mathbf{D}_n$ and relative depth maps $\mathbf{R}_n$. Table 4 summarizes the depth estimation results according to different combinations of ordinary and relative maps. The following observations can be made:

- When only a single ordinary depth map is used, a higher resolution one provides better results.
- Relative depths maps should be combined with at least one ordinary map to reconstruct depths, since they do not contain scale information (*i.e*. the mean depth). However, relative maps are more effective than ordinary ones. For example, when all relative maps are combined with the lowest resolution $\mathbf{D}_3$, the RMSE (lin) score is 0.538, which is better than that ($= 0.550$) of combining all ordinary maps.
- Combining $\mathbf{D}_3$ with four relative maps $\mathbf{R}_3$, $\mathbf{R}_4$, $\mathbf{R}_5$, $\mathbf{R}_6$ provides comparable or even better performances than using all ten depth maps. For example, the former yields $\rho = 0.914$, while the latter $\rho = 0.912$. This indicates that the additional ordinary maps rather distort the ground-truth depth ordering of a scene. Thus, we use only the five depth maps ($\mathbf{D}_3$, $\mathbf{R}_3$, $\mathbf{R}_4$, $\mathbf{R}_5$, $\mathbf{R}_6$) in the default mode.
- Adding $\mathbf{R}_7$ to the default mode improves the performances only slightly.

More ablation studies and more experimental results are available in the supplemental document.

5. Conclusions

We proposed a novel approach to monocular depth estimation, which uses relative depth maps. First, we developed the encoder-decoder network that has multiple decoder blocks for estimating relative depths, as well as ordinary ones, at various scales. To reduce complexity, we restored an entire relative depth map from selectively estimated data using the ALS algorithm. Finally, we reconstructed an optimal depth map through the depth map decomposition and the depth component combination. Experiments demonstrated that the proposed algorithm provides the state-of-the-art performance, and an ablation study showed that relative depth maps are more effective than ordinary ones in preserving the depth ordering of a scene.

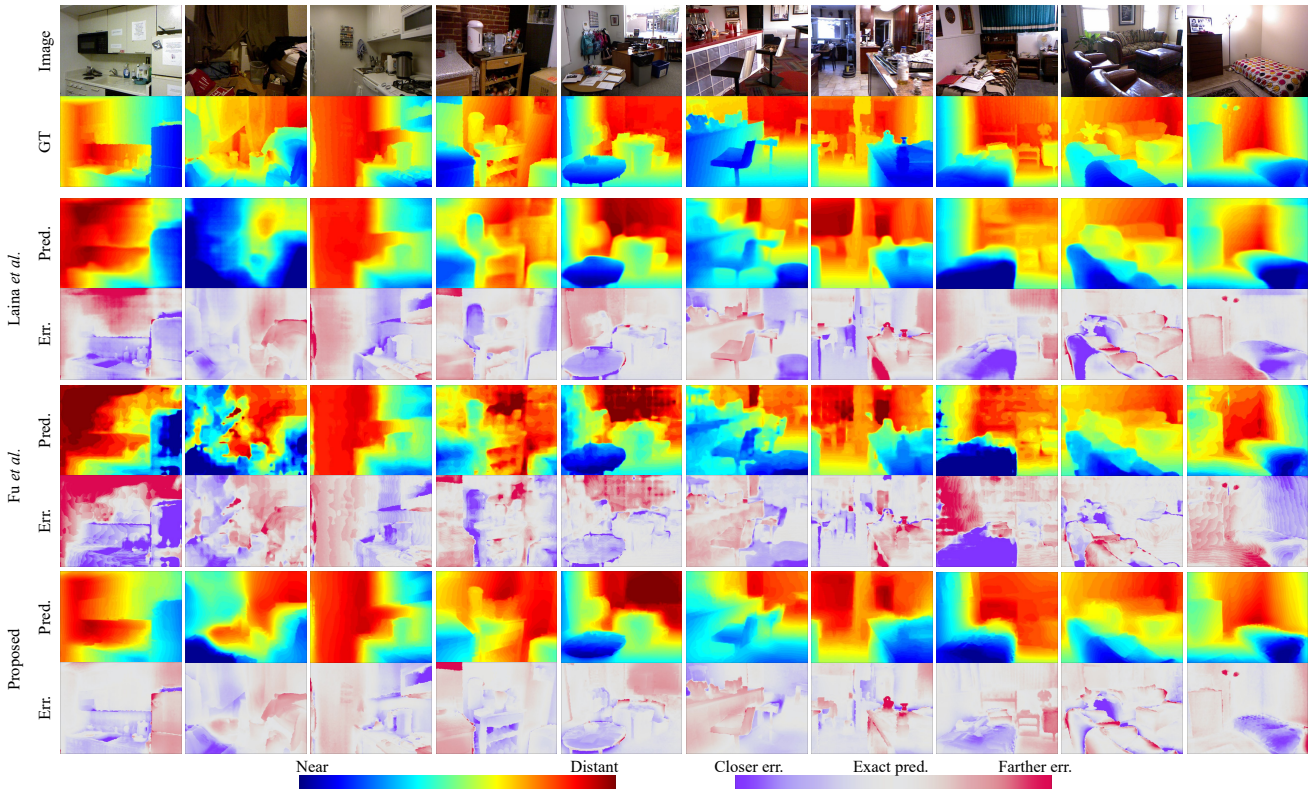

Figure 5. Qualitative comparison of Laina *et al*. [31], Fu *et al*. [13], and the proposed algoirhtm. Predicted depth maps (Pred), and error maps (Err) of relative depths are provided for easier comparison.

Figure 6. Qualitative comparison of depth map 3D visualization results of Laina *et al*. [31], Fu *et al*. [13], and the proposed algoirhtm.

Acknowledgments

This work was supported in part by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-2016-0-00464) supervised by the IITP(Institute for Information & communications Technology Promotion) and in part by the National Research Foundation of Korea (NRF) through the Korea Government (MSIP) under Grant NRF-2018R1A2B3003896.

References

[1] A. Altman and J. Gondzio. Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization. *Optimization Methods and Software*, 11(1-4):275–302, Jan. 1999.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, Dec. 2017.

[3] M. Baig and L. Torresani. Coupled depth learning. In *WACV*, 2016.

[4] J. Biswas and M. Veloso. Depth camera based indoor mobile robot localization and navigation. In *ICRA*, 2012.

[5] Avrim Blum, John Hopcroft, and Ravindran Kannan. Foundations of Data Science. 2015.

[6] A. Chakrabarti, J. Shao, and G. Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. In *NIPS*, 2016.

[7] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *NIPS*, 2016.

[8] C. M. Cheng, S. J. Lin, S. H. Lai, and J. C. Yang. Improved novel view synthesis from depth image with large baseline. In *ICPR*, 2008.

[9] E. Delage, H. Lee, and A. Y. Ng. A dynamic Bayesian network model for autonomous 3D reconstruction from a single indoor image. In *CVPR*, 2006.

[10] Persi Diaconis and R. L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):262–268, 1977.

[11] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[12] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[13] H. Fu, M Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.*, 32(11):1231–1237, Sept. 2013.

[15] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers Norwell, 1991.

[16] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010.

[17] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2341–2353, Dec. 2011.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[20] M. Heo, J. Lee, K. R. Kim, and C. S. Kim. Monocular depth estimation using whole strip masking and reliability-based refinement. In *ECCV*, 2018.

[21] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge, 2 edition, 2012.

[22] G. Huang, Y. Li, and G. Pleiss. Snapshot ensembles: Train 1, get m for free. In *ICLR*, 2017.

[23] G. Huang, Z. Liu, and L. van der Maaten. Densely connected convolutional networks. In *CVPR*, 2017.

[24] G. Pollastri J. Cheng, Z. Wang. A neural network approach to ordinal regression. In *IEEE International Joint Conference on Neural Networks*, 2008.

[25] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(11):2144–2158, Oct. 2014.

[26] R. H. Keshavan, S. Oh, and A. Montanari. Matrix completion from a few entries. In *IEEE International Symposium on Information Theory*, 2009.

[27] S. Kim, K. Park, K. Sohn, and S. Lin. Unified depth prediction and intrinsic image decomposition from a single image via joint convolutional neural fields. In *ECCV*, 2016.

[28] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 8:30–37, Aug. 2009.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[30] A. Kundu, Y. Li, F. Daellert, F. Li, and J. M Rehg. Joint semantic segmentation and 3D reconstruction from monocular video. In *ECCV*, 2014.

[31] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016.

[32] D. C. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NIPS*, 2010.

[33] J. H. Lee, M. Heo, K. R. Kim, and C. S. Kim. Single-image depth estimation based on fourier domain analysis. In *CVPR*, 2018.

[34] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *Int. J. Robot. Res.*, 34(4-5):705–724, Apr. 2015.

[35] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, Aug. 2004.

[36] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In *CVPR*, 2015.

[37] L. Li and H. T. Lin. Ordinal regression by extended binary classification. In *NIPS*, 2007.

[38] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *CVPR*, 2010.

[39] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015.

[40] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2024–2039, Oct. 2016.

[41] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *CVPR*, 2014.

[42] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theory*, 28(2):129–137, Mar. 1982.

[43] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[44] P. Ndjiki-Nya, M. Köppel, D. Doshkov, H. Lakshman, P. Merkle, K. Müller, and T. Wiegand. Depth image-based rendering with advanced texture synthesis for 3-D video. *IEEE Trans. Multimedia*, 13(3):453–465, June 2011.

[45] Y. Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k2)$. *Soviet Mathematics Doklady*, 27(2):372–376, Feb. 1983.

[46] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua. Ordinal regression with multiple output cnn for age estimation. In *CVPR*, 2016.

[47] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *CVPR*, 2018.

[48] A.N. Rajagopalan, S. Chaudhuri, and U. Mudenagudi. Depth estimation and image restoration using defocused stereo pairs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(11):1521–1525, Nov. 2004.

[49] B. Recht. A simpler approach to matrix completion. *The Journal of Machine Learning Research*, 12:3413–3430, Dec. 2011.

[50] X. Ren, L. Bo, and D. Fox. RGB-D scene labeling: Features and algorithms. In *CVPR*, 2012.

[51] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, 2016.

[52] T. L. Saaty. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15(3):234–281, June 1977.

[53] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS*, 2005.

[54] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3-D scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):824–840, Oct. 2009.

[55] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.*, 47:7–42, Apr. 2002.

[56] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Commun. ACM*, 56(1):116–124, Jan. 2013.

[57] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012.

[58] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Densely connected convolutional networks. In *AAAI*, 2017.

[59] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[60] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for oneshot human pose estimation. In *CVPR*, 2012.

[61] 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 *CVPR*, 2015.

[62] A. Wedel, U. Franke, J. Klappstein, T. Brox, and D. Cremers. Realtime depth estimation and obstacle detection from monocular video. In *Joint Pattern Recognition Symposium*, 2006.

[63] 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 *CVPR*, 2018.

[64] D. Xu, W. Ouyang, X. Wang, and N. Sebe. PAD-Net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, 2018.

[65] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation. In *CVPR*, 2017.

[66] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*, 2018.

[67] J. Yang, B. Price, and S. Cohen. Object contour detection with a fully convolutional encoder-decoder network. In *CVPR*, 2016.

[68] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys. Accurate 3D pose estimation from a single depth image. In *ICCV*, 2011.

[69] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018.

[70] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman. Learning ordinal relationships for mid-level vision. In *ICCV*, 2015.