Neural Processing Letters (2021) 53:1205–1228
https://doi.org/10.1007/s11063-021-10437-6

Perceptual Monocular Depth Estimation

Janice Pan[1] · **Alan C. Bovik**[1]

Accepted: 27 January 2021 / Published online: 10 February 2021

Abstract
Monocular depth estimation (MDE), which is the task of using a single image to predict scene depths, has gained considerable interest, in large part owing to the popularity of applying deep learning methods to solve "computer vision problems". Monocular cues provide sufficient data for humans to instantaneously extract an understanding of scene geometries and relative depths, which is evidence of both the processing power of the human visual system and the predictive power of the monocular data. However, developing computational models to predict depth from monocular images remains challenging. Hand-designed MDE features do not perform particularly well, and even current "deep" models are still evolving. Here we propose a novel approach that uses perceptually-relevant natural scene statistics (NSS) features to predict depths from monocular images in a simple, scale-agnostic way that is competitive with state-of-the-art systems. While the statistics of natural photographic images have been successfully used in a variety of image and video processing, analysis, and quality assessment tasks, they have never been applied in a predictive end-to-end deep-learning model for monocular depth. Correspondingly, no previous work has explicitly incorporated perceptual features in a monocular depth-prediction approach. Here we accomplish this by developing a new closed-form bivariate model of image luminances and use features extracted from this model and from other NSS models to drive a novel deep learning framework for predicting depth given a single image.

Keywords Monocular depth estimation · Natural scene statistics · Depth estimation · Bivariate natural scene statistics · Neural networks

1 Introduction

The human visual system (HVS) uses both binocular and monocular cues to perceive depths and instantaneously reconstruct the 3D world. Even from a single image, without binocular cues, humans are able to easily gain a good understanding of the 3D geometry of the scene, e.g., relative distances and object sizes. Recently, modeling depth from monocular luminance

✉ Janice Pan
janicepan@utexas.edu

Alan C. Bovik
bovik@ece.utexas.edu

[1] Laboratory for Image and Video Engineering (LIVE), The University of Texas at Austin, Austin, USA

has gained interest, and deep learning approaches to depth estimation have shown promising results. However, designing computational models to estimate depth from monocular images, or Monocular Depth Estimation (MDE), remains an ill-posed, challenging problem.

Recent work in estimating depth from monocular images has made use of deep network architectures and training on large databases [6,8,11,13,14,26,28–30,63]. Some models were trained with stereo or multiple views [13,14,26], but the amount of multi-view ground-truth data that is available is limited. Other approaches require post-processing, refinement steps, or multi-phase training [8,29] or use networks having tens of millions of trained parameters [6,11,28,30,63]. One previous approach used natural image statistic features in a Bayesian framework to regress depth using a simpler learner (Support Vector Regressor) [56], however that model requires training two distinct sub-models, and depth-prediction is subsequently a two-part process. Here we show that perceptually available information provided by the local, scale-invariant statistics of natural scenes, or natural scene statistics (NSS), can be used to predict depth in a simple end-to-end network, yielding significantly improved results.

Natural Scene Statistics (NSS) have proven to be useful in understanding the evolution of the HVS and for exploring complex visual problems [5,39,47,57,61]. In a related study, the authors of Liu et al. [32] deployed statistical models of luminance/chrominance and depth/disparity NSS, and the relationships between them, to drive a Bayesian stereo algorithm to predict disparity. The authors of Su et al. [51] extended that work by introducing NSS models of the marginal and conditional distributions of luminances/chrominances and depths/disparities of natural scenes and used them to improve a chromatic Bayesian stereo algorithm. More recently, bivariate and correlation NSS models for natural images and depth maps were developed, but these only operate on spatially adjacent bandpass responses [52,53]. NSS models of spatially adjacent luminance coefficients have also been shown to provide strong features on visual problems [33,34], but they have yet to be applied to the problem of depth prediction method, monocular, stereo, or otherwise.

Here we advance the problem of perceptual depth prediction, by first introducing a new closed-form correlation model of local bivariate luminance statistics. We then use features derived from this model, along with other univariate and bivariate NSS features, to train a deep neural network (DNN) to predict inverse-range patches from luminance patches. When building this model, we represent each inverse-range patch using a sparse code, each element of which supplies a weight on each patch within a dictionary of image patches, which was learned from patches sampled from depth perception databases of ground-truth luminance-depth images. We discuss the construction of our network and emphasize connections and parallels to perceptual models and concepts. To our knowledge, no previous work has derived or made explicit use of perceptually-relevant features when predicting depth from monocular images. In the sections following, we explain the new correlation NSS model, the dictionary construction, and the network design, including the data processing steps, the network architecture, the training and testing details, and the results. We show that our simple network trained using perceptually-relevant features and a perceptually-relevant loss function is able to predict dense depth maps in a manner that is competitive with state-of-the-art methods.

2 Related Work

2.1 NSS

Extensive work has been done analyzing and modeling the natural statistics of 2D and 3D scenes and how the HVS processes this information. In particular, many statistical models

have been developed that make use of statistical models of bandpass responses of luminance and depth/disparity of real-world images [9,32,40,44,51,56,60]. Using locally normalized *luminance coefficients* when modeling such relationships has been explored to a lesser extent, though NSS computed in this domain are also perceptually relevant [33,34].

There exists a strong relationship between luminance and co-located depth in natural scenes [40]. Statistical models have been established that reliably capture the univariate [32,51] and bivariate [52,53] statistics of real-world photographs quite well. There are a variety of useful established closed-form NSS models of bandpass images [10,47,48,54,55], and recent work using them for the monocular depth estimation problem has shown promising results [56]. However, a bivariate model of luminance statistics as a function of orientation and separation between pixels would better capture the relationships between pixels and scene geometry. We develop and introduce a closed-form bivariate model for luminance coefficients in Sect. 3.

2.2 Monocular Depth Estimation

Estimating dense depth maps of naturalistic scenes is useful for many computer vision tasks, such as scene reconstruction and object detection/recognition. Estimating scene depths given only a single monocular image is a much more challenging problem than multi-view (stereo) or multi-frame (video) depth estimation. The latter two problems have been studied extensively [17,22,25,37,41,64], however the problem of monocular depth estimation is a much more poorly-posed problem and progress towards developing a solution has been more difficult.

Recent approaches to MDE that utilize DNNs have shown promising results [6,8,11,24,28–30,43,63]. However, some of the highest-performing methods limit the resolution of the output [8], require post-processing (correction) steps [6], require multi-phase training [63], or reformulate the problem as a variation of a classification task [6,11]. Some recent methods for MDE rely on stereo or multi-view data for training [12–14], thus potentially limiting their generalizability.

Instead of predicting depth, we predict the inverse of depth, which is proportional to disparity, and which we refer to as *inverse range*. The task of predicting the inverse of depth follows the objective of stereo correspondence methods. Stereo algorithms aim to estimate disparity, which can be used to compute depth directly if the baseline and focal length of the stereo cameras are known. Our method uses a novel modular DNN architecture that relies solely on monocular training data. It can be trained end-to-end, and because it operates patch-wise, it is able to produce predictions for images of any resolution.

3 Luminance and Depth NSS

One hypothesis of vision science is that the sensory systems have become, through evolution and adaptation, matched to the statistical properties of the signals to which they are exposed [1,42,59]. When attempting to solve the MDE problem, the only available data from which to predict depth is from single-view pixel data. Thus, we begin by exploring the statistical relationships that exist between the bandpass luminance coefficients of images and their corresponding depths. We develop a model that is able to capture the correlations between spatially neighboring luminance values across multiple scales. We apply the bivariate model introduced in Su et al. [52], [53] to the spatial domain, complete the correlation model, and

present it in a closed form. We then use features derived from the new correlation model, along with established univariate and bivariate luminance NSS features, to train a DNN to predict depths from monocular image patches.

3.1 Univariate NSS Model

In the 1980s, Ruderman [44] observed interesting outcomes that arose by applying a simple process of image modification. After digitizing a large set of outdoor photographs, he found that subtracting estimates of the local mean luminances from the pictures, then further processing by dividing by the estimates of local variance (an example of a 'divisive normalization transformation', or DNT), has a decorrelating and gaussianizing effect on the image data. The DNT process is perceptually significant and is a reasonable approximation to the nonlinear behavior of retino-cortical neurons [60]. We will start by reviewing simple spatial domain NSS models, beginning with the univariate model detailed in Mittal et al. [33,34]. Given an image I, define the normalized luminance values:

$$\hat{I}(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + C}, \tag{1}$$

where i, j index row and column pixels, respectively, C is a normalization constant that stabilizes the quotient when σ is small, and

$$\mu(i,j) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} w_{k,l} I_{k,l}(i,j), \tag{2}$$

and

$$\sigma(i,j) = \sqrt{\sum_{k=-K}^{K} \sum_{l=-L}^{L} w_{k,l} (I_{k,l}(i,j) - \mu(i,j))^2}, \tag{3}$$

where, as in Mittal et al. [33], we use a patch sampling window $w = \{w_{k,l} | k = -K, ..., K, l = -L, ..., L\}$ that is a 2D circularly/symmetric Gaussian weighting function sampled out to 3 standard deviations and rescaled to unit volume. In the experiments, we use $K = L = 3$, but operate over multiple scales.

Although in principle, the empirical distributions of the DNT luminance coefficients should be Gaussian-shaped, we instead fit the histograms with the zero-mean generalized Gaussian distribution (GGD) function:

$$f(x; \alpha, \sigma^2) = \frac{\alpha}{2\beta\Gamma(1/\alpha)} \exp\left(- \left(\frac{|x|}{\beta} \right)^{\alpha} \right), \tag{4}$$

where

$$\beta = \sigma \sqrt{\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}}, \tag{5}$$

and

$$\Gamma(a) = \int_0^{\infty} t^{a-1} e^{-t} dt, \quad a > 0. \tag{6}$$

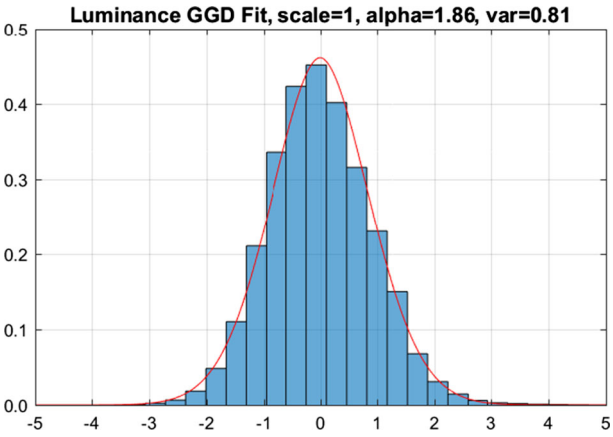


Fig. 1 GGD fit to bandpass, divisively normalized luminance histogram from the LIVE SV database [37].

The GGD is parameterized by α and σ^2, which represent the shape and variance, respectively, of the distribution. We use the maximum-likelihood method detailed in Sharifi and Leon-Garcia [45] to estimate α and σ^2. One of the databases that we use is the large new LIVE Surround-View (SV) Database [37], which contains over 130, 000 pairs of synthetically-generated naturalistic color images and ground-truth depth maps. We have found that the parameters of the best GGD fits to the DNT luminance coefficients of LIVE SV support the use of natural picture statistics models on the images in the LIVE SV database, as shown in Fig. 1: the distributions of the normalized luminance coefficients strongly tend towards a unit normal Gaussian [33,44].

The LIVE SV database was originally generated for the purpose of exploring surround-view driver assistance imaging systems, where the feeds from four fisheye cameras placed around a vehicle are combined to generate a top-down view of the vehicle and its surroundings. The SV database, contains fisheye stereo pairs and their corresponding depth maps, also subjected to the fisheye distortion. The spatial distortion can be removed to obtain typical rectilinear photographic images of city scenes, which include pedestrians, bikes, motorcycles, a variety of vehicles, and other realistic content. Instructions to download the database can be found at https://github.com/janicepan/live-sv/.

3.2 Bivariate NSS Model

Our bivariate NSS model extends the work done in Su et al. [52], [53], which explored the NSS of spatially adjacent directional bandpass (wavelet) responses. We use a simpler isotropic bandpass/DNT model. As with Su et al., [52], [53], we use a bivariate generalized Gaussian distribution (BGGD) to model the pairwise statistics of pixels having four relative orientations, separated by distance d, as shown in Fig. 2: the horizontal (0^o), right diagonal (45^o), vertical (90^o), and left diagonal (135^o).

The bivariate generalized Gaussian distribution is:

$$f(\mathbf{x}; \mathbf{M}, \alpha_b, \beta_b) = \frac{1}{|\mathbf{M}|} g_{\alpha_b, \beta_b}(\mathbf{x}^\top \mathbf{M}^{-1} \mathbf{x}), \tag{7}$$

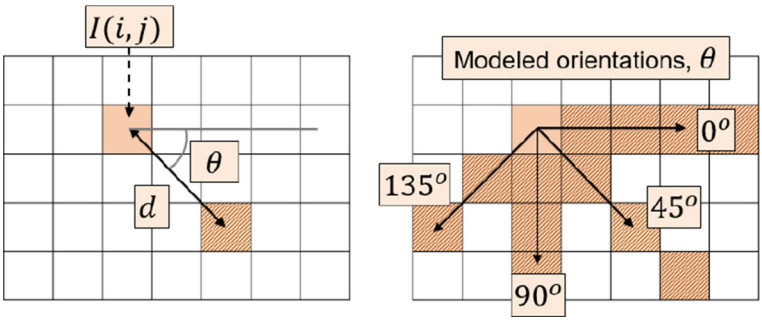


Fig. 2 Visualization of distances and orientations between pairs of pixels on which we model bivariate statistics. Left: distance is measured in pixels, while orientation is measured clockwise from the positive horizontal. Right: the four orientations considered in our bivariate statistical model.

where $\mathbf{x} \in \mathbb{R}^2$, $\mathbf{M}$ is a 2×2 covariance matrix, α_b and β_b are scale and shape parameters, respectively, and $g_{\alpha_b,\beta_b}(\cdot)$ is:

$$g_{\alpha_b,\beta_b}(y) = \frac{\beta_b \Gamma(1)}{\left(2^{\frac{1}{\beta_b}} \pi \alpha_b\right)^{\frac{1}{2}} \Gamma\left(\frac{N}{2\beta_b}\right)} e^{-\frac{1}{2}\left(\frac{y}{\alpha_b}\right)^{\beta_b}}, \quad y \in \mathbb{R}^+. \tag{8}$$

The elements of matrix $\mathbf{M}$ are pairwise covariances, for which we seek a simple closed-form correlation model so that the overall second-order statistical model (7), (8) is also of closed form. Figures 3 and 4 plot the correlations against the distance between the pixels for four different orientations, on two different databases. Figure 3 was generated on the LIVE Color+3D Database Release 2 [56], which contains 98 sets of high-definition (1920×1080) stereo color image pairs with co-registered dense ground-truth depth maps. The images in this database are pristine, natural images, i.e., with minimal distortion. The depth data was captured using a RIEGL VZ-400 terrestrial range scanner calibrated to the camera.

As previously mentioned, we also utilized the LIVE Surround-View (SV) Database [37], because of its size, which greatly facilitates the training of deep models, and the guaranteed accuracy of its ground-truth range measurements. Large amounts of ground-truth co-registered luminance and depth data are very difficult to obtain, and this dataset contains over 130, 000 pairs of naturalistic color images and corresponding depth maps. Since the data was generated synthetically, it contains perfectly calibrated co/registered luminance-/depth data. The correlation plots generated using this synthetic data nicely agree with those computed on pristine images captured by high-definition cameras, which, along with the univariate GGD fits (Fig. 1), supports the use of LIVE SV for developing MDE models of natural photographic + depth images.

We have found that a difference-of-exponentials can be used to accurately model the empirical correlation values:

$$\rho(d) = 2\mathrm{e}^{ad} - \mathrm{e}^{abd}, \tag{9}$$

where d is the pixel separation, a controls the decay steepness (and thus, the location of the dip that is evident in the plots), and b controls the width of the dip (or lack thereof). Figure 5 shows possible shapes the model (9) can exhibit when varying the parameters a and b. In the left plot, a is fixed at -0.5, while b varies, and in the right plot, b is fixed at 0.5, while a varies.

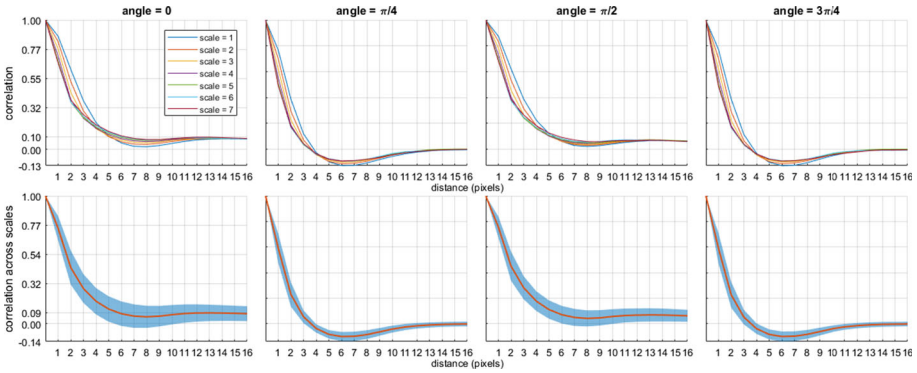


Fig. 3 Empirical luminance correlations computed on the LIVE Color+3D Release 2 Database. Top row: correlation as a function of distance for (from L to R) $\theta = \{0, 45^o, 90^o, 135^o\}$ and 7 different scales. Bottom row: correlation functions averaged over scales showing confidence bands.

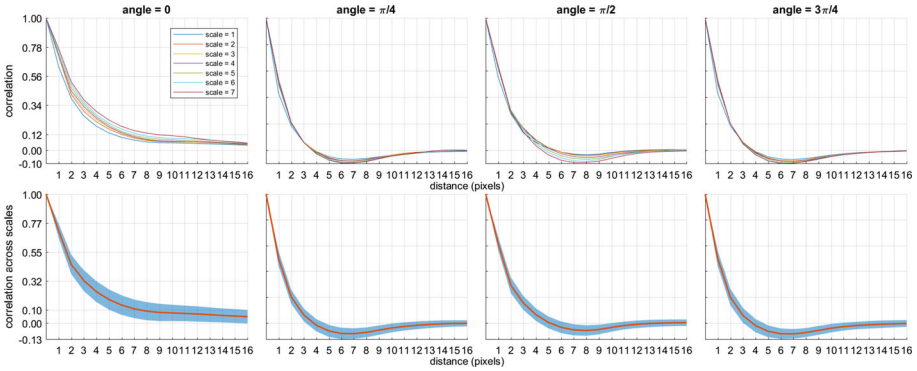


Fig. 4 Empirical luminance correlations computed on the LIVE SV Database. Top row: correlation as a function of distance for (from L to R) $\theta = \{0, 45^o, 90^o, 135^o\}$ and 7 different scales. Bottom row: correlation functions averaged over scales showing confidence bands.

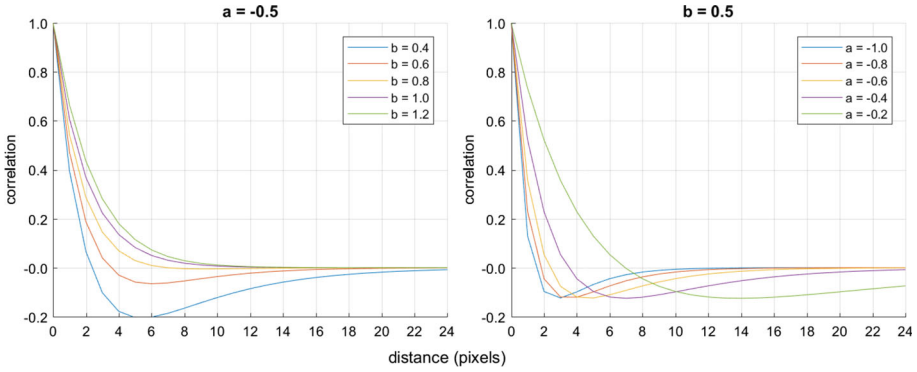


Fig. 5 Difference of exponentials (correlation model (9)) with (left) fixed a and varying b and (right) fixed b and varying a.

4 Sparse Dictionary Representation for Inverse-Range Maps

It is strongly believed that the HVS extracts statistical elements of visual stimuli to produce efficient representations [1,42,59]. Early on, Barlow [3] proposed that early visual processing in the brain acted to remove much of the considerable redundancy present in visual signals by generating statistically independent neural responses. Further, a significant amount of prior work has demonstrated the connection between NSS and sparse representations [2,9,35,36]. In particular, Field [9] showed that simple cells in the primary visual cortex (V1) create sparse representations of the natural world, which is largely made up of regular structures of 3D objects and surfaces. Following Field's paper, there has been extensive work showing that receptive fields similar to those of simple cells can be optimized to produce sparse representations of natural scenes [4,18,20,35,58]. Regularities in depth are strongly tied to luminance regularities, and depth and luminance statistics can be similarly modeled [21,31,32,49–53].

Su *et. al.* [56] postulated that depth maps tend to be composed of simple, regular patterns, and used a small dictionary of five canonical patterns to represent priors in a Bayesian depth prediction framework. They achieved promising results on predicting dense depth maps. We adopt a similar approach as Su et al. [56], but with important differences. We use a significantly larger collection of picture + depth data to learn a larger, more descriptive dictionary of inverse-range patterns. Instead of using steerable pyramids, we build dictionaries of divisively-normalized inverse-range patches, computed by applying (1) to the inverse-range maps:

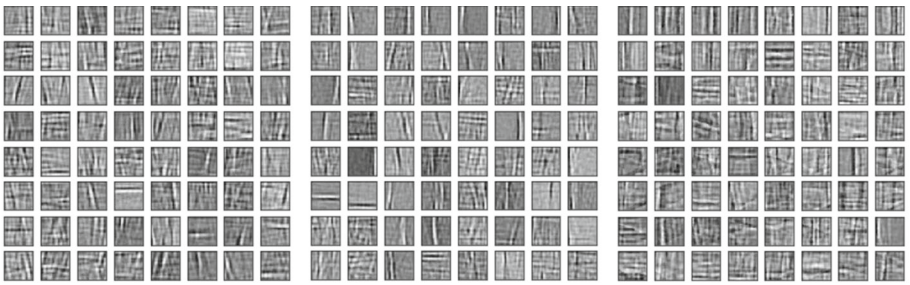
$$\hat{I}_{ir}(i,j) = \frac{I_{ir}(i,j) - \mu_{ir}(i,j)}{\sigma_{ir}(i,j) + C_{ir}}. \tag{10}$$

On each database that we used to develop and test our models, we randomly extracted 200,000 DNT inverse-range patches of size 31×31 across a range of scales. We first scaled each image to multiple resolutions, the choice of which we discuss in Sect. 5.2, and for each image, regardless of resolution, we extracted patches of constant size 31×31. We then applied least angle regression (LARS) [7], which is a stepwise regression algorithm that, at each step, selects a patch from a collection of possible dictionary elements that has the strongest correlation with the data, and then iteratively selects patches based upon a refitting of residuals to build the dictionary. The candidates for the dictionary patches were initialized using singular value decomposition (SVD) to be the orthogonal basis for $\mathbb{R}^{961}$ (because $31^2 = 961$). In other words, if we have $n \times p$ data matrix A, where n is the number of image patches used to learn the dictionary (200,000), and $p = 31^2 = 961$, then applying SVD gives:

$$\mathbf{A} = \mathbf{USV^T}, \tag{11}$$

and the candidate dictionary elements are the rows of $\mathbf{SV^T}$. By iteratively selecting the strongest patch to include in the dictionary, we obtain a small subset of orthogonal patches that can be used to reconstruct any DNT inverse-range image, thereby removing a significant amount of redundancy in the representation of the input data. LARS is implemented in Python in the `MiniBatchDictionaryLearning` function from the decomposition module within the scikit-learn library [38]. By including patches from multiple scales during construction of the dictionary, we were able to build a reasonably scale-agnostic model.

Using this method, we obtained a set of 64 31×31-sized basis patches, so that any image patch within the database can be represented using a linear combination of these 64 bases.



(a) LIVE SV (Pan et al., 2018) (b) LIVE Color+3D (Su et al., 2017) (c) NYU Depth V2 (Silberman et al., 2012)

Fig. 6 Dictionaries of learned sparse inverse-range patches computed on each of the three tested databases.

Each patch from each image in the database can thus, instead of being represented by 31^2 pixels, be represented by a sparse 64-length code, each element of which represents a weight for the corresponding dictionary patch. Figure 6 shows the learned dictionaries of inverse-range patches on each of the three datasets we tested. We also tried learning dictionaries of different sizes (36 and 81), and found that the smaller 36-element dictionary yielded image reconstructions with significantly greater loss, whereas the 64- and 81-element dictionaries yielded comparable high-accuracy image reconstructions.

Figure 7 shows examples of reconstructions of one DNT inverse-range image from each database. To demonstrate the applicability of the dictionaries to images of different scales, the examples in Fig. 7 are of varying resolutions. The example from the SV database in Fig. 7a is 360×640, which is 25% of the resolution of the original (720×1280); the example from the LIVE Color+3D Database in Fig. 7b is 270×480, or $1/16$-th the original resolution of 1080×1920; and the example from the NYU Depth Dataset in Fig. 7c is of full resolution 480×640 (460×620 with some invalid edge pixels removed).

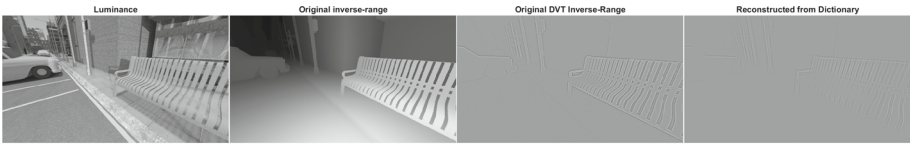
5 Method

We now explain the components of our inverse-range patch prediction model. We start by describing the model input features in Sect. 5.1. Then we describe each of the relevant luminance + depth datasets, and how we processed each of them to extract the necessary features in Sect. 5.2, We detail the design of the network architecture in Sect. 5.3, and relate how we trained the model in Sect. 5.4.

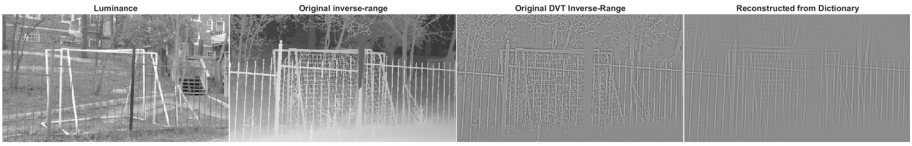
5.1 NSS Features

The goal is to take an input luminance patch that is a projection of part of a 3D scene, and densely predict its corresponding depth patch. We use patches of size 31×31 throughout. From each patch within a luminance image, we extract a 982-length feature vector $\boldsymbol{F}$ comprising:

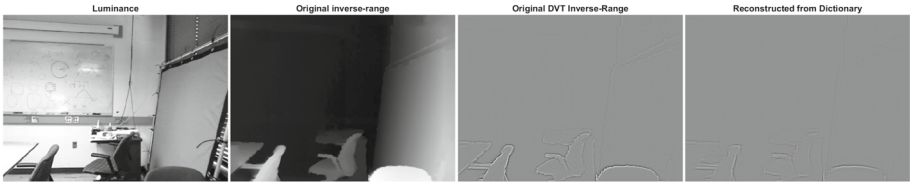
- DNT luminance coefficients: L_{ij} for $i, j \in 1, ..., 31$,
- Luminance patch mean: μ_L,
- Luminance patch variance: v_L,



(a) Example from LIVE SV (Pan et al., 2018)



(b) Example from LIVE Color+3D (Su et al., 2017)



(c) Example from NYU Depth V2 (Silberman et al., 2012)

Fig. 7 From left to right in each row: Original luminance image; ground-truth inverse-range map; divisively normalized bandpass ground-truth inverse-range; divisively normalized bandpass inverse-range image reconstructed from the dictionary elements in Fig. 6a

- Normalized y-coordinate of the luminance patch: y,
- Luminance GGD parameters: $\alpha_{\text{GGD}}, \sigma_{\text{GGD}}$,
- Luminance BGGD parameters: $\alpha_{\text{BGGD}_k}, \beta_{\text{BGGD}_k}$ for $k = [1, ..., 4]$,
- Luminance correlation model parameters: a_k, b_k for $k = [1, ..., 4]$,

where k indexes the four pairwise orientations $\{0^o, 45^o, 90^o, 135^o\}$, and μ_L and v_L are extracted during the divisive normalization step. The GGD parameters are estimated using the moment-matching approach used in Mittal et al. [33] and proposed in Sharifi and Leon-Garcia [45]; the BGGD parameters are estimated using the moment-matching approach for fitting multivariate generalized Gaussian distributions (MGGD) detailed in Gómez et al. [16]; and the correlation model parameters are estimated using the Nelder-Mead simplex algorithm [27], as implemented in the `fminsearch` MATLAB function, to fit the constrained difference of exponentials (9) to the empirical correlations, for integer pixel separations ranging from $d = 1$ to $d = 16$ pixels.

5.2 Datasets

Next we describe the datasets that we used and explain how the data was processed for both training and testing. On each dataset, we extract and use only the luminance channel, and compute inverse-range as the reciprocal of depth. We also selected 20 images from each dataset that we set aside to evaluate the performance of the trained inverse-range prediction network on full images. These images were not included when generating patch data to train and test the network. We also list in Table 1 a few database features and processing choices. The variable C_{ir} represents the normalization constant used in (1) for inverse-range images, which is required to build the inverse-range dictionary specific to each dataset and to extract

Table 1 Database Features and Processing

DB	Res.	Scales	C_{ir}	Depths
LIVE SV	720×1280	7	0.1	<50m
LIVE Color+3D	1080×1920	7	0.0001	–
NYU Depth V2	480×640	5	0.001	–

the sparse code for generating the training and testing data. The choice of C_{ir} is different for each database, because it depends on the variances of the luminance patches within the databases, which arises from various factors, such as the precision of the data, the complexity of the scenes, and any inherent noise in the data. We manually chose C_{ir} for each database to not overpower the variance in the DNT equation denominator.

Additionally, because we aim to build a scale-agnostic model, we extracted a range of scales of images within each database depending on the original resolution of images in that database. The images in the NYU dataset [46], for instance, are native 480×640, which is much smaller than the native images in both the LIVE SV [37] and LIVE Color+3D [56] databases, so we only extract five scales (including the original) on NYU, as opposed to the seven which we extract from the others.

LIVE SV Database The LIVE SV Database [37] is a large synthetic database with over 130,000 co-registered pairs of color images and ground-truth depth maps. Although all of the images in the database were captured with a fisheye lens, the intrinsic camera parameters are known, so we transformed all the images and depth maps to their rectilinear forms. To generate the training and testing data, we randomly sampled patches from 500 luminance-depth image pairs. In this database, some pixels in the sky or on complex object surfaces (e.g., tree leaves) have infinite ground-truth depth recorded, so they were excluded from the training and testing sets, and only depths less than 50 meters were estimated.

LIVE Color+3D Database Release 2 The LIVE Color+3D Database Release 2 [56] contains 98 sets of stereo color image pairs with co-registered depth maps of outdoor scenes. Because the left and right images and depths in each stereo pair are quite similar in nature, we randomly selected 20 left images for final evaluation and excluded their corresponding right images from both the training and testing data.

NYU Depth Dataset V2 The NYU Depth Dataset V2 [46] contains 1449 pairs of aligned color and depth maps of indoor scenes. Due to occlusions and measurement limitations, the raw depth maps contain missing values. These missing depth measurements are often in key spatial locations, as they correspond to changes in object boundaries and sharp changes in depth. The authors provide interpolated measurements for missing values, and because it is important that our model be able to learn such geometries in a scene, we used these dense maps to extract training and testing data.

5.3 NN Architecture

In order to optimize our predictions of inverse-range using NSS features, we built a modular network, motivated by our normalization transformation of the inverse-range images and NSS. If we consider any inverse-range map as being a reconstruction from its DNT form, as in (10), then we may formulate the inverse-range-prediction problem as a combination of four

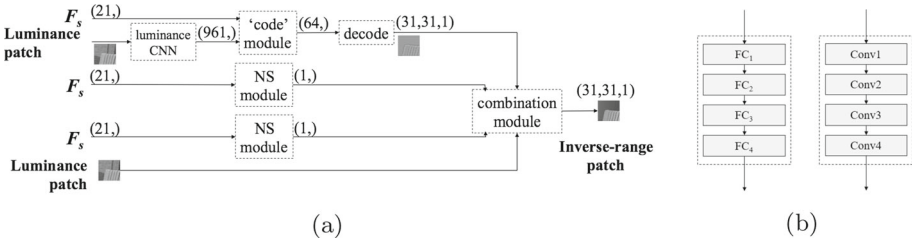


Fig. 8 **a** Network flow diagram, in which the four-layer fully-connected architecture shown on the left in (**b**) is used in the code-prediction, mean-prediction, and variance-prediction modules, and the four-layer convolutional architecture shown on the right in (**b**) is used in the combination module as well as to extract luminance features to input into the code-, mean-, and variance-prediction modules. The two blocks labeled *NS module* in (**a**) are the mean- and variance-prediction modules.

smaller problems: first, predicting the DNT patch, the patch mean, and the patch variance, and then combining those three predictions in a reverse divisive-normalization transformation step. We model our network on this formulation, as depicted in the network flow diagram in Fig. 8a. One module, the 'code' block, is designed to predict a 64-length feature vector, which is intended to represent the code used to reconstruct the DNT inverse-range patch from dictionary patch elements. To obtain the inverse-range values from the DNT patch, we also require the patch mean and variance. However, instead of training separate networks to predict mean and variance separately, we use two normalizing scalar (NS) blocks in our network to predict scalar features to be passed to the combination module to predict the final inverse-range patch.

We show the specific architectures of the modules in Fig. 8b. The code and NS modules use the four fully-connected layers, while the combination module uses the four convolutional layers. The luminance blocks also use the four convolutional layers with an additional fully-connected layer to obtain the intermediate code and scalar features of the appropriate dimensions. We provide details of all layer dimensions for our highest-performing network in Table 2, and note that our network contains approximately 880K parameters, which is orders of magnitude smaller than other high-performing models [11,28,30].

We tested module architectures of different depths and widths and tried extracting luminance feature vectors of different lengths. We also tested fully-connected architectures, fully/convolutional architectures, and other combinations besides the one represented in Table 2, along with options of including residual connections. The chosen output length of the luminance feature/extraction block is 961, as displayed in Fig. 8a, which indicates that the luminance CNN extracts a feature vector of the same dimensions as the input patch. We found that extracting shorter feature vectors compromised performance. Our final network also does not use any skip or residual connections. We did not find that adding them between fully-connected layers improved performance, and we obtained similar results using an adapted version of ResNet [19] as the combination module, however that network is more complex and takes much longer to train and execute on test data. Importantly, our proposed network requires far fewer parameters than previous high-performing networks designed and trained for monocular depth estimation. Much of the predictive power of the system lies in the use of perceptually relevant features.

If the features listed in Sect. 5.1 are denoted as F, then F_s is a subset of F that excludes the DNT luminance coefficients. For the NS blocks, the only inputs are F_s, whereas, for the 'code'-prediction module, the DNT luminance patch is first passed through four convolutional

Table 2 Network Architecture Details

Module	Input Dimensions	Output Dimensions	Layer Dimensions			
			FC_1	FC_2	FC_3	FC_4
Code	(982,)	(64,)	524	262	131	131
NS 'mean'	(21,)	(1,)	11	6	3	3
NS 'variance'	(21,)	(1,)	11	6	3	3
			$Conv_1$	$Conv_2$	$Conv_3$	$Conv_4$
Luminance	(31,31,1)	(961,)	(3,3,4)	(3,3,8)	(3,3,16)	(3,3,32)
Combination	(31,31,4)	(31,31,1)				

layers to extract a 961-length feature vector before being concatenated with F_s to form the input feature vector of length 982. We also conducted experiments where we excluded the NSS features and only trained on luminance patches. In these experiments, the input to each of the three modules was a 961-length luminance feature vector, extracted using a separate CNN (each having the same architecture as the luminance CNN detailed in Table 2) for each module. We found that the results suffered significantly without including any NSS features in the model.

The network begins with these three independent branches to represent the components in the reverse divisive normalization process (10), because any image patch can be reconstructed from its DNT, mean, and variance. The 64-length vector resulting from the code-prediction module represents the sparse code, which is used to reconstruct the DNT patch from the dictionary elements. This step is represented by the 'decode' block in Fig. 8a, which takes a 64-length inverse-range patch code as input and reconstructs an estimate of the DNT inverse-range patch, which is then combined with the outputs of the NS modules in a combination module, yielding the estimate of the final inverse-range patch.

5.4 Training

To train the network, which we did for each database separately, we randomly sampled 625,000 luminance-depth patch pairs. We reserved 20% (125,000 pairs) for testing, and used the remaining 80% (500,000 pairs) for training. During the training process, 10% of the training patches were reserved for validation.

We evaluated a number of loss functions, including the mean squared error (MSE), mean absolute error (MAE), difference of structural similarity (DSSIM), Wang et al. [62]; Keras-contrib [23], and weighted combinations of the three. The non/traditional loss, DSSIM, is based on the Structural SIMilarity index (SSIM), which considers luminance, contrast, and structural similarities between two images, taking a maximum of 1 when the images being compared are identical. To use this index in a loss function, we therefore subtract its value from 1, and in the Keras implementation we used, the DSSIM loss is clipped at 0.5. We found that the MAE preserved finer details, which is to be expected, while a combination loss (MAE-DSSIM) equally weighing MAE and DSSIM produced the best results.

We trained the network end-to-end, while also utilizing losses at the outputs of the 'code' and NS modules. For the first half of training, we used a combination loss comprising the overall MAE-DSSIM inverse-range loss as well as three intermediate layer MAE losses:

- the MAE loss between the output of the 'code'-prediction block and the ground-truth sparse code of the DNT inverse-range patch;

- the MAE loss between the output of one of the NS modules and the ground-truth inverse-range patch mean;
- and the MAE loss between the output of the other NS module and the ground-truth inverse-range patch variance.

The weight ratio between the overall loss and each of the three intermediate losses was $1 : 0.01$. The validation loss was computed at the end of each epoch. We used a patience threshold of 10 epochs to determine when to change or stop training, meaning that we monitor the validation loss to determine when it no longer improved (i.e., the validation loss decreased) for 10 consecutive epochs. The first time this event happened, the weights for the intermediate losses were dropped, and only the overall MAE-DSSIM inverse-range patch loss was used for the rest of training, which stopped when we no longer saw improvement in the validation loss for another 10 consecutive epochs. The final model was taken to be the one having the lowest validation loss.

We used this two-tier training scheme to initially gently guide the 'code' and NS modules to predict the quantities required for patch reconstruction from DNT components (10). We removed these guiding losses before finishing training, which we noticed had a small impact on the visual results, but systematically improved the quantitative results. We also tried training end-to-end with only the MAE-DSSIM loss, and found that the network sometimes struggled more to accurately predict the inverse-range values of object interiors.

5.4.1 Data Augmentation

We also observed that the inverse-range patch variances of natural images are heavily skewed toward zero, because scene depths are predominantly smooth and continuous with discontinuities mainly corresponding to object edges. To avoid biasing our model to predicting inverse-range patches having zero variance, we augmented the training data by sampling the patches based on their variances, and by combining these samples with the training samples. Specifically, we computed a sample weight for each training patch to be its variance over the sum of all training patch variances. We then sampled, with replacement, 500,000 patches, i.e., the same number of patches as in the training set, appended these patches to the original training data, and randomly selected 500,000 patches from this augmented set for training.

In Fig. 9, we show, for a random sample of LIVE SV patches, the histogram of inverse-range patch variances on the left, where the severity of the bias toward zero is apparent. After resampling, the distribution is still skewed toward zero, but the bias is not as severe, as seen in the histogram on the right in Fig. 9. By implementing this resampling approach, our training data distribution is still significantly representative of the true data distribution, but we place a greater emphasis on learning inverse-range values on patches having higher inverse-range variance, which are more likely to correspond to points of interest or object edges. We trained networks with and without this resampling step and found that the predictions resulting from using the resampled training data to be both quantitatively and qualitatively better.

6 Results

While the final models are trained to predict inverse-range in a patchwise manner, the true test of each model is how well it can predict dense inverse-range maps on entire images. To evaluate the final models, we used the 20 randomly-selected **evaluation** images from each database that did not contribute any training or testing patches to build the model.

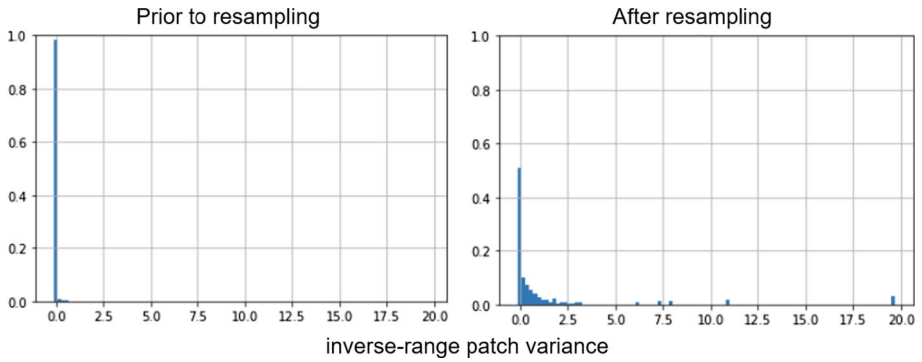


Fig. 9 Histograms of inverse-range patch variances: (left) before resampling and (right) after resampling using the method detailed in Sect. 5.4.1. Plots are shown for a randomly sampled 100,000 patches from the LIVE SV data, but we observed similar trends on the other databases as well.

As previously mentioned, if the evaluation image was part of a stereo pair from the LIVE Color+3D database, the corresponding image also made no contribution to the sets of training and testing patches, because the image content between stereo image pairs is almost always nearly identical.

To evaluate performance, we compute the mean absolute error (MAE) and the normalized mean absolute error (nMAE), which is the MAE divided by the ground truth values:

$$\text{nMAE}(G, P) = \frac{|G - P|}{|G|}, \tag{12}$$

where G and P are the ground-truth and predicted inverse-range maps, respectively.

In Table 3, we show the average and median of each error metric across the 20 evaluation images for each database when the model was trained on patches extracted from images of the same resolution. In other words, for each scale, 650,000 training and testing patches were extracted only from images of that scale. Thus, the scales and image resolutions listed in Table 3 correspond to both the model and evaluation data, and the model trained for each scale can be considered to be *scale-aware*. The results in Table 4, on the other hand, were obtained from models trained on patches extracted across all scales, i.e., 7 scales on both of the LIVE databases and 5 scales on the NYU dataset. These models can therefore be considered as *scale-agnostic*, and the scale and image resolutions listed in Table 4 correspond only to the evaluation data for which the results are listed, because a single multi-scale model was trained for each database.

The results reported in Tables 3 and 4 are errors, so smaller numbers indicate better performance. We considered the normalized MAE when comparing model performances, since it weighs errors based on the ground truth values. Therefore, major differences between depth distributions in scenes from different databases are normalized. Generally, the multi-scale models, which were trained on data across all scales, performed slightly worse than scale-specific models trained on scale-specific data. However, the difference was small, and the multi-scale results strongly support efficacy of using the scale invariant NSS to learn scale-agnostic networks. We also observed that for any given database, models trained on higher resolution data did not necessarily perform better in terms of nMAE. Similarly, models trained on data at different scales also did not necessarily perform better on higher resolution data. In fact, there was no obvious relationship between resolution and performance, which supports the generalizability of the model and features.

Table 3 Results from Single-Scale Models

Database	Scale Index	Image Resolution	Average MAE	Median MAE	Average nMAE	Median nMAE
LIVE SV	0	720 × 1280	0.4251	0.3736	0.3347	0.3221
	1	509 × 905	0.4166	0.4050	0.3176	0.3014
	2	360 × 640	0.3986	0.3739	0.3204	0.3174
	3	255 × 453	0.4094	0.3750	0.3431	0.3470
	4	180 × 320	0.3820	0.3559	0.3096	0.3190
	5	127 × 226	0.4016	0.3543	0.3328	0.3191
	6	90 × 160	0.3804	0.3610	0.2934	0.2941
LIVE Color+3D	0	1080 × 1920	0.0432	0.0394	0.6060	0.5207
	1	764 × 1358	0.0414	0.0352	0.5470	0.4916
	2	540 × 960	0.0379	0.0314	0.5277	0.4627
	3	382 × 678	0.0396	0.0319	0.5394	0.4186
	4	270 × 480	0.0439	0.0310	0.5309	0.4693
	5	191 × 339	0.0453	0.0308	0.5489	0.4809
	6	135 × 240	0.0453	0.0342	0.5304	0.4553
NYU Depth V2	0	460 × 620	0.1500	0.1085	0.3094	0.3083
	1	325 × 438	0.1460	0.1074	0.3002	0.2951
	2	230 × 310	0.1402	0.1061	0.2894	0.2898
	3	163 × 219	0.1343	0.1033	0.2814	0.2810
	4	115 × 155	0.1260	0.1035	0.2697	0.2790

We also compared our model's performance on the test images of each database with the performance of the models presented by Laina et. al [28] and Godard et. al [15]. Although another model [11] has been reported to provide better performance than the Laina model, we could not reproduce the reported results, and the authors of Fu et al. [11] do not provide training code to do so. In Fu et al., [11], the second best performance on the NYU dataset was obtained using Laina, and since the work in Fu et al. [11] was published, Godard et al. [15] has been reported to outperform other models, so we used Laina and Godard as benchmarks to evaluate our model's performance. Both the Laina and Godard models are trained on full images and predicts complete depth maps, although Laina's predicted depth maps are at a lower resolution. Thus, for each database, we formed the training data using the images from which we extracted patches to train our patch-based model.

For the Laina model, we also resized the LIVE SV and LIVE Color+3D images to 480×640 to avoid modifying the original model. Additionally, because the model was initially presented to predict range, we trained and evaluated the Laina model on range images and took the inverse of the predictions for the sake of comparison. We tried training the models to predict inverse-range directly, but this approach yielded worse results. The Godard model requires image dimensions to be multiples of 32, so we cropped the images in the LIVE SV, LIVE Color+3D, and NYU Depth V2 datasets as minimally as possible to obtain valid dimensions. We show the performance results of both models along with those from our own scale-agnostic model in Table 5.

We show example predictions computed by both our scale-aware and scale-agnostic models, as well as the predictions produced by the Laina [28] and Godard [15] models in Figs. 10,

Table 4 Results from Multi-Scale Models

Database	Scale Index	Image Resolution	Average MAE	Median MAE	Average nMAE	Median nMAE
LIVE SV	0	720 × 1280	0.4642	0.4341	0.3745	0.3642
	1	509 × 905	0.4471	0.4228	0.3757	0.3646
	2	360 × 640	0.4423	0.4100	0.3663	0.3668
	3	255 × 453	0.4474	0.4167	0.3537	0.3522
	4	180 × 320	0.4439	0.4263	0.3345	0.3387
	5	127 × 226	0.4643	0.4181	0.3228	0.3152
	6	90 × 160	0.4669	0.4671	0.3093	0.2904
LIVE Color+3D	0	1080 × 1920	0.0497	0.0408	0.7224	0.6231
	1	764 × 1358	0.0518	0.0387	0.6617	0.5737
	2	540 × 960	0.0488	0.0340	0.6491	0.5653
	3	382 × 678	0.0461	0.0321	0.6165	0.5274
	4	270 × 480	0.0459	0.0323	0.5963	0.5042
	5	191 × 339	0.0449	0.0341	0.5903	0.5150
	6	135 × 240	0.0437	0.0337	0.5778	0.5050
NYU Depth V2	0	460 × 620	0.1510	0.1107	0.3029	0.3035
	1	325 × 438	0.1464	0.1087	0.2967	0.2953
	2	230 × 310	0.1433	0.1077	0.2948	0.2886
	3	163 × 219	0.1383	0.1037	0.2877	0.2934
	4	115 × 155	0.1328	0.0966	0.2773	0.2816

Table 5 Comparison with Previous Models

Database	Model	MSE Mean	MSE Median	MAE Mean	MAE Median	nMAE Mean	nMAE Median
LIVE SV	Pan	**0.554**	**0.349**	**0.464**	0.434	**0.375**	**0.364**
	Laina	5.868	5.337	1.938	1.914	0.677	0.667
	Godard	2.038	0.460	0.590	**0.203**	0.587	0.545
LIVE Color+3D	Pan	**0.005**	**0.003**	**0.050**	**0.041**	**0.722**	**0.623**
	Laina	0.494	0.478	0.672	0.660	10.176	9.764
	Godard	2.8395	0.633	0.997	0.658	0.724	0.672
NYU Depth V2	Pan	0.047	0.022	0.151	**0.111**	**0.303**	0.304
	Laina	**0.036**	**0.021**	**0.146**	0.123	0.336	0.326
	Godard	1.217	0.527	0.756	0.622	0.321	**0.216**

Bolded values indicate best performance

11, and 12. On each database, we chose predictions from two different scales of our models to demonstrate the generalizability of our scale-specific models to be scale-agnostic. (Refer to Table 3 or 4 for scale resolutions.) The Laina model performed quite well on the NYU dataset, however, our models outperformed the Laina model with respect to normalized MAE. The complexities contained in the LIVE SV Database and the insufficient amount of training data in the LIVE Color+3D Database likely contributed to the poor visual results delivered by the Laina model. The Godard model performed well, both qualitatively and in terms of MAE and

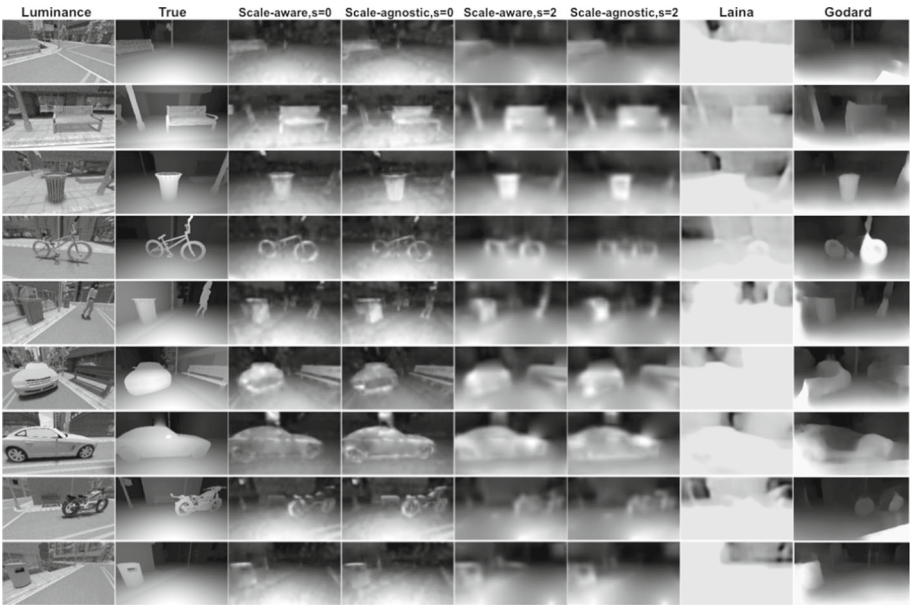


Fig. 10 Inverse-range prediction results on the LIVE SV Database.

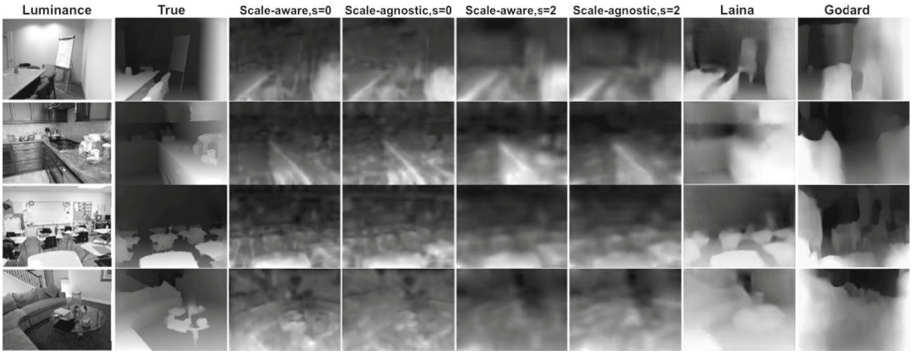


Fig. 11 Inverse-range prediction results on the LIVE Color+3D Database.

nMAE, on all three datasets, and achieved the lowest median MAE and nMAE, however, out models were comparable in both metrics and considerably outperformed the Godard model in terms of MSE and average MAE and nMAE.

6.1 Module Outputs

As mentioned, the modular design of the network was motivated by the DNT deconstruction of the inverse-range map. The three components needed to reconstruct any inverse-range patch are the divisively-normalized patch, which can be reconstructed from the 64-length code, along with the patch mean and variance. The latter two are represented in the network by the NS blocks, which predict scalar outputs. Though training begins with weakly biasing these two NS modules to predict the inverse-range patch mean and variance, this bias is removed

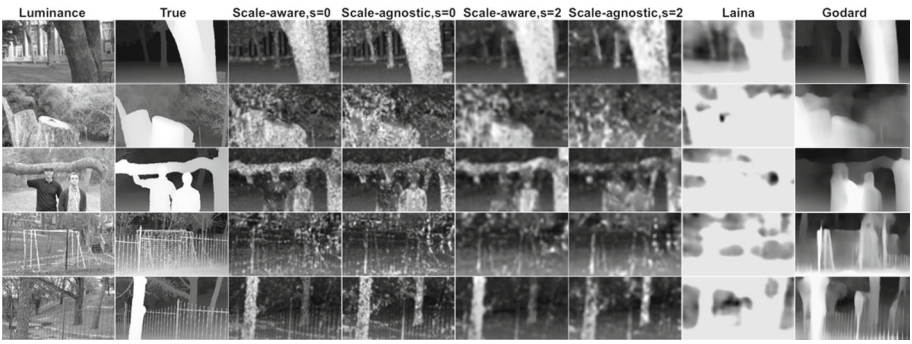


Fig. 12 Inverse-range prediction results on the NYU Depth Database.

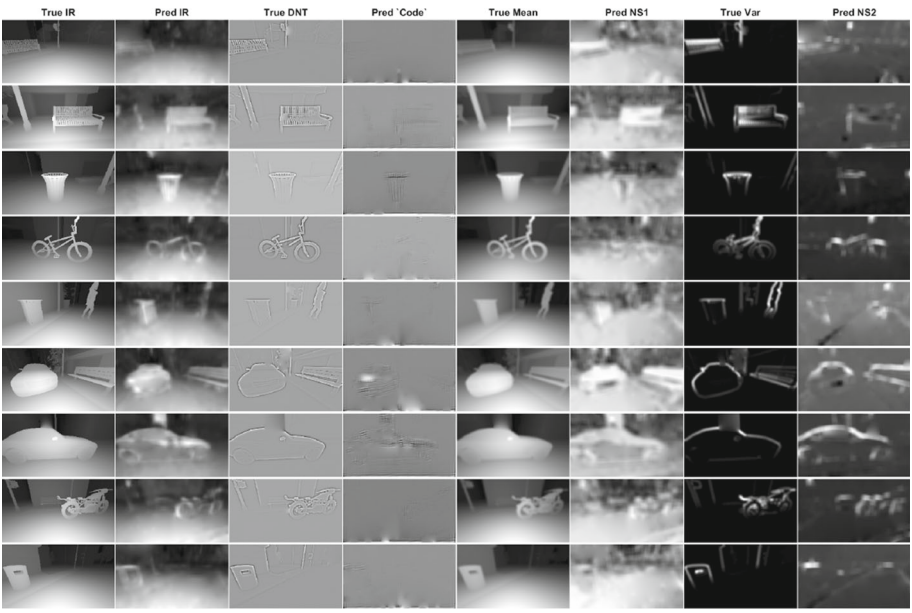


Fig. 13 Intermediate model outputs. From left to right: the ground-truth inverse-range image; the predicted inverse-range image; the ground-truth divisively-normalized inverse-range image; the predicted divisively-normalized inverse-range image; the ground-truth inverse-range mean image; the predicted inverse-range mean image; the ground-truth variance image; and the predicted variance image.

to finish training. We are still, however, able to demonstrate that the values predicted by these intermediate network layers closely resemble the intended mean and variances scalars. In Fig. 13, we show all three intermediate network outputs from the model trained on scale-2 data for all the LIVE SV images displayed in Fig. 10. The predicted DNT, mean, and variance images all display similar characteristics to their ground-truth maps, which further supports the HVS motivation behind the design of our network.

Fig. 14 Comparison of results for models trained on single-scale (scale 2) data for the SV database, both with (middle column) and without (last column) using NSS features as inputs.

Table 6 With and Without NSS

Database	Model	MSE		MAE		nMAE	
		Mean	Median	Mean	Median	Mean	Median
LIVE SV	With NSS	**0.427**	**0.332**	**0.399**	**0.374**	**0.320**	**0.317**
	Without NSS	1.016	0.821	0.709	0.655	0.631	0.651

Bolded values indicate best performance

6.2 The Value of NSS Features

As mentioned in Sect. 5.3, we also trained models without using the NSS features (F_s) as inputs to the model, as a way of probing the efficacy of these features. In these experiments, we excluded the NSS features and trained the models using only the luminance patches, which were fed into three separate CNN modules, each configured according to the luminance CNN described in Table 2. These luminance feature vectors were then separately fed into the same 'code' and 'NS' blocks shown in Fig. 8a. For the SV examples shown in Fig. 10, we also show, in Fig. 14, the difference between the predictions when NSS features were included as input features (middle column) and excluded entirely from the network (last column). These examples clearly illustrate the enhanced performance afforded by including NSS features in the monocular depth estimation training process. A quantitative comparison is also provided in Table 6 for the scale-specific models (trained on 1/4-resolution SV data) that produced the results in Fig. 14. Both the visual and numeric results strongly support the use of NSS features as inputs to the model, and the idea that these kinds of fundamental, perceptually relevant statistical features can be used to significantly improve learned MDE models.

7 Conclusion

In this work, we presented a new bivariate correlation model for image luminance coefficients, which we use to devise features extracted from it, along with other univariate and bivariate NSS features, in a modular patch-based network that is trained to predict inverse-range patches from a single luminance patch. We also utilized a learned dictionary to enable the representation of DNT inverse-range patches by sparse codes, which served as intermediary features within the predictive model. Our use of DNT inverse-range patches (as well as DNT luminance patches) was motivated by the low-level visual processing in the HVS, and representing inverse-range patches in such a way significantly reduced the required complexity of our network.

We trained our network on patches extracted from images of the same scale as well as on patches extracted from images across multiple scales to see how well our models could generalize. We found that our scale/agnostic models performed comparably to the scale-/specific models, likely because the features that are input to the model are based on NSS models, which are scale/invariant. We presented a novel perceptually/motivated deep-learning approach to densely estimating inverse-range maps, and showed that it can perform well on complicated scenes, as well as on databases lacking a sufficient number co-registered image/depth pairs to train deep full-image models. Our model is able to produce results competitive with current state-of-the-art monocular depth-estimation models and is the first approach to explicitly use perceptually-relevant features in a solution to the challenging problem of predicting depth from monocular images.

References

1. Attneave F (1954) Some informational aspects of visual perception. Psychol Rev 61(3):183
2. Barlow HB (1972) Single units and sensation: A neuron doctrine for perceptual psychology? Perception 1(4):371–394
3. Barlow HB et al (1961) Possible principles underlying the transformation of sensory messages. Sensory Commun 1:217–234
4. Bell AJ, Sejnowski TJ (1997) The "independent components" of natural scenes are edge filters. Vis Res 37(23):3327–3338
5. Bovik AC (2013) Automatic prediction of perceptual image and video quality. Proc IEEE 101(9):2008–2024
6. Cao Y, Wu Z, Shen C (2018) Estimating depth from monocular images as classification using deep fully convolutional residual networks. IEEE Trans Circuits Syst Video Technol 28(11):3174–3182
7. Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 32(2):407–499
8. Eigen D, Puhrsch C, Fergus R (2014) Depth map prediction from a single image using a multi-scale deep network. Adv Neural Inf Proc Syst 27:2366–2374
9. Field DJ (1987) Relations between the statistics of natural images and the response properties of cortical cells. J Opt Soc Am 4(12):2379–2394
10. Field DJ (1999) Wavelets, vision and the statistics of natural scenes. Philos Trans Royal Soc Lond Ser A Math Phys Eng Sci 357(1760):2527–2542
11. Fu H, Gong M, Wang C, Batmanghelich K, Tao D (2018) Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE conference on computer vision and pattern Recognition
12. Garg R, BG VK, Carneiro G, Reid I (2016) Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: Proc. Eur. Conf. Comput. Vis., pp 740–756
13. Godard C, Mac Aodha O, Brostow GJ (2017) Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE conference on computer vision and pattern Recognition
14. Godard C, Mac Aodha O, Firman M, Brostow G (2018) Digging into self-supervised monocular depth estimation. arXiv preprint arXiv:1806.01260
15. Godard C, Mac Aodha O, Firman M, Brostow GJ (2019) Digging into self-supervised monocular depth estimation. In: Proc. IEEE Int'l Conf. Comput. Vis., pp 3828–3838
16. Gómez E, Gomez-Viilegas M, Marin J (1998) A multivariate generalization of the power exponential family of distributions. Commun Stat Theory Methods 27(3):589–600
17. Ha H, Im S, Park J, Jeon HG, So Kweon I (2016) High-quality depth from uncalibrated small motion clip. In: Proceedings of the IEEE conference on computer vision and pattern Recognition
18. van Hateren JH, Ruderman DL (1998) Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. Proc Royal Soc Lond Ser B Biol Sci 265(1412):2315–2320
19. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern Recognition, pp 770–778
20. Hyvärinen A, Hoyer P (2000) Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. Neural Comput 12(7):1705–1720
21. Jou JY, Bovik AC (1989) Improved initial approximation and intensity-guided discontinuity detection in visible-surface reconstruction. Comput Vis Graphics Image Process 47(3):292–326
22. Karsch K, Liu C, Kang SB (2014) Depth transfer: Depth extraction from video using non-parametric sampling. IEEE Trans Pattern Anal Mach Intell 36(11):2144–2158
23. Keras-contrib (2018) DSSIM Loss Function. https://github.com/keras-team/keras-contrib/blob/master/keras_contrib/losses/dssim.py
24. Kim S, Park K, Sohn K, Lin S (2016) Unified depth prediction and intrinsic image decomposition from a single image via joint convolutional neural fields. In: Proc. Eur. Conf. Comput. Vis., pp 143–159
25. Kong N, Black MJ (2015) Intrinsic depth: Improving depth transfer with intrinsic images. In: IEEE Int'l Conf. Comput. Vis
26. Kuznietsov Y, Stuckler J, Leibe B (2017) Semi-supervised deep learning for monocular depth map prediction. In: Proceedings of the IEEE conference on computer vision and pattern Recognition
27. Lagarias JC, Reeds JA, Wright MH, Wright PE (1998) Convergence properties of the Nelder-Mead simplex method in low dimensions. SIAM J Optim 9(1):112–147
28. Laina I, Rupprecht C, Belagiannis V, Tombari F, Navab N (2016) Deeper depth prediction with fully convolutional residual networks. In: Int'l Conf. 3D Vision, pp 239–248
29. Li B, Shen C, Dai Y, Van Den Hengel A, He M (2015) Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In: Proceedings of the IEEE conference on computer vision and pattern Recognition, pp 1119–1127

30. Liu F, Shen C, Lin G, Reid I (2016) Learning depth from single monocular images using deep convolutional neural fields. IEEE Trans Pattern Anal Mach Intell 38(10):2024–2039
31. Liu Y, Cormack LK, Bovik AC (2009) Luminance, disparity, and range statistics in 3D natural scenes. In: Proc. SPIE, Human Vis. and Electronic Imaging, vol 7240, p 72401G
32. Liu Y, Cormack LK, Bovik AC (2011) Statistical modeling of 3D natural scenes with application to bayesian stereopsis. IEEE Trans Image Process 20(9):2515–2530
33. Mittal A, Moorthy AK, Bovik AC (2012) No-reference image quality assessment in the spatial domain. IEEE Trans Image Process 21(12):4695–4708
34. Mittal A, Soundararajan R, Bovik AC (2013) Making a "completely blind" image quality analyzer. IEEE Signal Process Lett 20(3):209–212
35. Olshausen BA, Field DJ (1997) Sparse coding with an overcomplete basis set: A strategy employed by v1? Vis Res 37(23):3311–3325
36. Olshausen BA, Field DJ (2004) Sparse coding of sensory inputs. Curr Opin Neurobiol 14(4):481–487
37. Pan J, Mueller M, Lahlou T, Bovik AC (2018) Orthogonally-divergent fisheye stereo. In: Int'l Conf. Advanced Concepts for Intell. Vis. Systems, pp 112–124
38. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
39. Portilla J, Strela V, Wainwright MJ, Simoncelli EP (2003) Image denoising using scale mixtures of gaussians in the wavelet domain. IEEE Trans Image Process 12(11):1338–1351
40. Potetz B, Lee TS (2003) Statistical correlations between two-dimensional images and three-dimensional structures in natural scenes. J Opt Soc Am 20(7):1292–1303
41. Rajagopalan A, Chaudhuri S, Mudenagudi U (2004) Depth estimation and image restoration using defocused stereo pairs. IEEE Trans Pattern Anal Mach Intell 26(11):1521–1525
42. Rao RP, Olshausen BA, Lewicki MS (2002) Probabilistic models of the brain: perception and neural function. MIT press, Cambridge
43. Roy A, Todorovic S (2016) Monocular depth estimation using neural regression forest. In: Proceedings of the IEEE conference on computer vision and pattern Recognition
44. Ruderman DL (1994) The statistics of natural images. Netw Comput Neural Syst 5(4):517–548
45. Sharifi K, Leon-Garcia A (1995) Estimation of shape parameter for generalized gaussian distributions in subband decompositions of video. IEEE Trans Circuits Syst Video Technol 5(1):52–56
46. Silberman N, Hoiem D, Kohli P, Fergus R (2012) Indoor segmentation and support inference from RGBD images. In: Eur. Conf. Comput. Vis., pp 746–760
47. Simoncelli EP, Olshausen BA (2001) Natural image statistics and neural representation. Ann Rev Neurosci 24(1):1193–1216
48. Sinno Z, Caramanis C, Bovik AC (2018) Towards a closed form second-order natural scene statistics model. IEEE Trans Image Process 27(7):3194–3209
49. Su CC, Bovik AC, Cormack LK (2011) Natural scene statistics of color and range. In: IEEE Int'l Conf. Image Process., pp 257–260
50. Su CC, Bovik AC, Cormack LK (2012) Statistical model of color and disparity with application to bayesian stereopsis. In: IEEE Southwest Symp. Image Anal. and Interpretation, pp 169–172
51. Su CC, Cormack LK, Bovik AC (2013) Color and depth priors in natural images. IEEE Trans Image Process 22(6):2259–2274
52. Su CC, Cormack LK, Bovik AC (2014) Bivariate statistical modeling of color and range in natural scenes. In: Proc. SPIE, Human Vis. and Electronic Imaging, vol 9014. https://doi.org/10.1117/12.2036505
53. Su CC, Cormack LK, Bovik AC (2014) New bivariate statistical model of natural image correlations. In: IEEE Int'l Conf. Acoustics, Speech, Signal Process., pp 5362–5366
54. Su CC, Cormack LK, Bovik AC (2015a) Closed-form correlation model of oriented bandpass natural images. IEEE Signal Process Lett 22(1):21–25
55. Su CC, Cormack LK, Bovik AC (2015b) Oriented correlation models of distorted natural images with application to natural stereopair quality evaluation. IEEE Trans Image Process 24(5):1685–1699
56. Su CC, Cormack LK, Bovik AC (2017) Bayesian depth estimation from monocular natural images. J Vis 17:22–22
57. Tang H, Joshi N, Kapoor A (2011) Learning a blind measure of perceptual image quality. In: Proceedings of the IEEE conference on computer vision and pattern Recognition, pp 305–312
58. Van Hateren JH, van der Schaaf A (1998) Independent component filters of natural images compared with simple cells in primary visual cortex. Proc Royal Soc Lond Ser B Biol Sci 265(1394):359–366
59. Wainwright MJ, Schwartz O (2002) Natural image statistics and divisive. Probabilistic Models of the Brain: Perception and Neural Function, Chap 10, p 203

60. Wainwright MJ, Schwartz O, Simoncelli EP (2001) Natural image statistics and divisive normalization: Modeling nonlinearities and adaptation in cortical neurons. Perception and Neural Function, Probabilistic Models of the Brain. MIT Press, pp 203–222
61. Wang Z, Bovik AC (2011) Reduced- and no-reference image quality assessment. IEEE Signal Process Mag 28(6):29–40
62. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP et al (2004) Image quality assessment: From error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
63. Xu D, Ricci E, Ouyang W, Wang X, Sebe N (2017) Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In: Proceedings of the IEEE conference on computer vision and pattern Recognition, pp 5354–5362
64. Zagoruyko S, Komodakis N (2015) Learning to compare image patches via convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern Recognition