Depth Estimation by Parameter Transfer With a Lightweight Model for Single Still Images

Hongwei Qin, *Student Member, IEEE*, Xiu Li, *Member, IEEE*, Yangang Wang, Yongbing Zhang, *Member, IEEE*, and Qionghai Dai, *Senior Member, IEEE*

***Abstract*— In this paper, we propose a novel method for automatic depth estimation from color images using parameter transfer. By modeling the correlation between color images and their depth maps with a set of parameters, we get a database of parameter sets. Given an input image, we extract the high-level features to find the best matched image sets from the database. Then the set of parameters corresponding to the best match are used to estimate the depth of the input image. Compared with the past learning-based methods, our trained model consists only of trained features and parameter sets, which occupy little space. We evaluate our depth estimation method on several benchmark RGB-D (RGB + depth) data sets. The experimental results are comparable to the state-of-the-art results, while the model size is very small and very suitable for mobile devices, demonstrating the promising performance of our proposed method.**

***Index Terms*— 3D reconstruction, depth estimation, parameter transfer.**

I. Introduction

IMAGES captured with conventional cameras lose the depth information of the scene. However, scene depth is of great importance for many computer vision tasks. 3D applications, like 3D reconstruction for scenes (e.g., Street View on Google Maps), robot navigation, 3D videos, and free-view video [1], [2], all rely on scene depth. Depth information can also be useful for 2D applications, such as image enhancing [3] and scene recognition [4]. Recent RGB-D imaging devices like Kinect are greatly limited to the perceptive range and depth resolution. Neither can extract depth for the existing 2D images. Therefore, depth estimation from color images has been a useful research subject.

In this paper, we propose a novel depth estimation method to generate depth maps from single still images. Our method applies to arbitrary color images. We build the connection between image and depth with a set of parameters. A parameter set database is constructed and the parameter sets are transferred to input images to get the corresponding depth maps. Some estimation results are shown in Fig. 1.

As a reminder, this paper is organized as follows. In Section II, the related techniques are surveyed. In Section III, we introduce our proposed depth estimation by parameter transfer (DEPT) method in detail. We demonstrate our method on the RGB-D benchmark data sets in Section IV. Finally, we conclude this paper in Section V.

Manuscript received October 1, 2015; revised January 11, 2016; accepted June 10, 2016. Date of publication June 13, 2016; date of current version April 3, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 71171121, in part by the National 863 High Technology Research and Development Program of China under Grant 2012AA09A408, and in part by the Shenzhen Science and Technology Project under Grant JCYJ20151117173236192 and Grant CXZZ20140902110505864. This paper was recommended by Associate Editor K. Mu Lee.

H. Qin, X. Li, and Y. Zhang are with the Department of Automation, Tsinghua University, Beijing 100084, China, and also with the Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China (e-mail: li.xiu@sz.tsinghua.edu.cn).

Y. Wang is with Microsoft Research Asia, Beijing 100080, China.

Q. Dai is with the Department of Automation, Tsinghua University, Beijing 100084, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2016.2580438

II. Related Works

In this section, we introduce the techniques related to this paper, which are, respectively, depth estimation from a single image and parameter transfer.

A. Depth Estimation From Single Images

The reason why depth estimation from a single image is possible lies in that there are some monocular depth cues in a 2D image. Some of these cues are inferred from local properties like color, shading, haze, defocus, texture variations and gradients, and occlusions. Global cues are also crucial to inferring depth, as the ability humans have. Therefore, integrating local and global cues of a single image to estimate depth is reasonable.

There are semiautomatic and automatic methods for depth estimation from single images. Horry *et al.* [5] propose *tour into the picture*, whereby the user interactively adds planes to an image to make animation. The work of Zhang *et al.* [6] requires the user to add constraints manually to images to estimate depth.

Automatic methods for single image depth estimation come up in recent years. Hoiem *et al.* [7] propose *automatic photo popup*, which reconstructs an outdoor image using assumed planar surfaces of it. Delage *et al.* [8] develop a Bayesian framework applied to indoor scenes. Saxena *et al.* [9] propose a supervised learning approach, using a discriminatively trained Markov random field that incorporates multiscale local and global image features. Then, they improve this method in [10]. After that, depth estimation from predicted semantic labels is proposed by Liu *et al.* [11]. A more sophisticated model called feedback-enabled cascaded classification model is proposed by Li *et al.* [12]. One typical depth estimation method is depth transfer, developed by Karsch *et al.* [13]. This method first builds large-scale RGB-D images and features database and then acquires the depth of the input image by

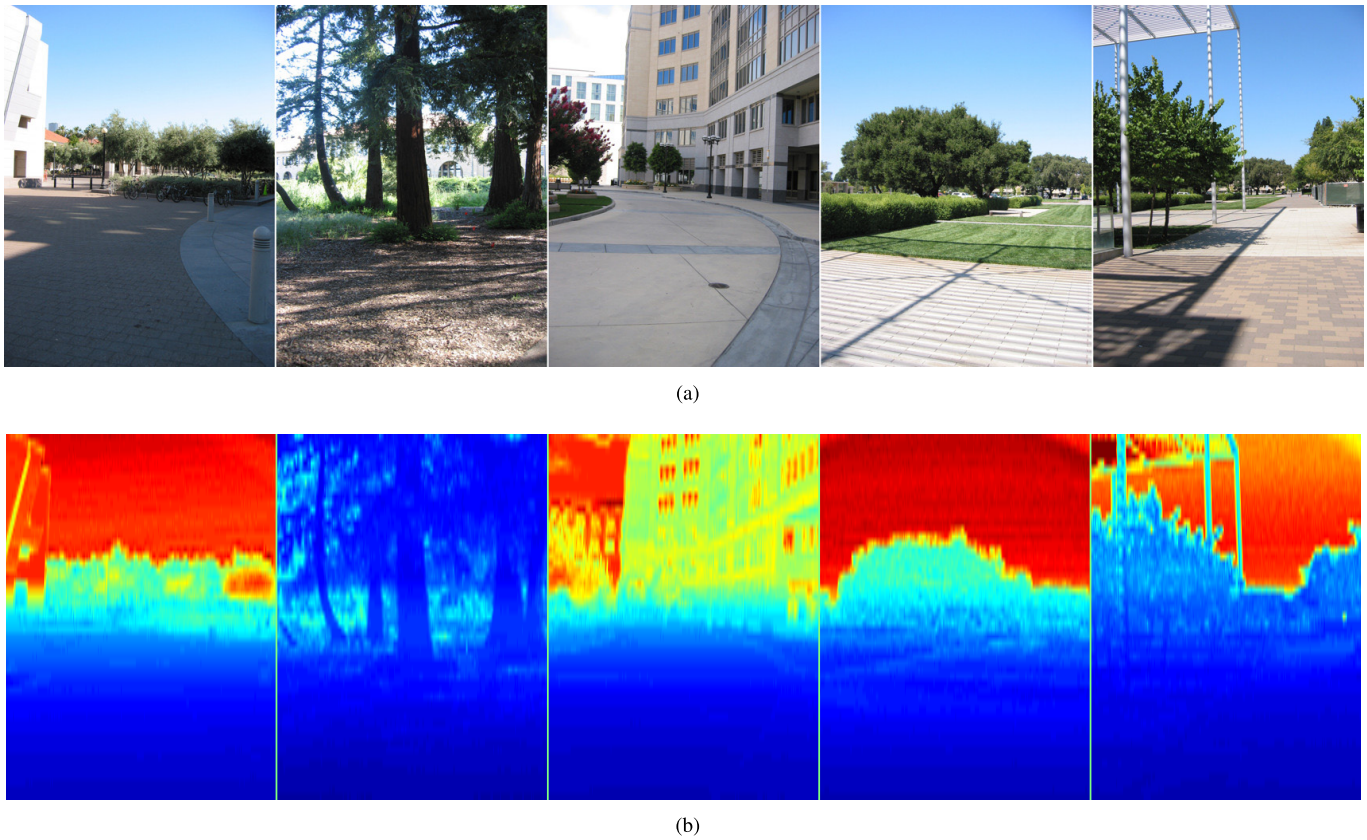


Fig. 1. Selected images and corresponding depth maps estimated by DEPT. The darker the red is, the further (from the imaging device) the objects are. The darker the blue is, the closer the objects are. (a) Test images. (b) Estimated depth maps by DEPT.

transferring the depth of several similar images after warping and optimizing procedures.

There are several recent works that try to solve the depth estimation problem and semantic segmentation problem unitedly. Ladicky *et al.* [14] propose to predict pixelwise semantic class labels to improve both depth estimation and semantic segmentation performance. Eigen *et al.* [15], [16] use a multiscale convolutional architecture to refine local depth prediction with global information. Wang *et al.* [17] propose to decompose the image into local segments for region-level depth and semantic prediction under the guidance of global layout.

Besides, there are several other efforts on depth estimation with unified global and local information. Liu *et al.* [18] use continuous variables encoding the depth of the superpixels in the input image and discrete variables representing relationships between neighboring superpixels to perform inference through a graphical model. Zhuo *et al.* [19] propose to use a hierarchical representation of the indoor scene and refine the depth map guided by global layout. Liu *et al.* [20] propose a method to refine depth map predicted by convolutional networks by continuous conditional random field. Baig *et al.* [21], [22] express the global depth map of an image as a linear combination of a depth basis learned from examples. The basis is actually a dictionary of the training data set and the images near the cluster centroids are picked as basis elements. Our concurrent and independent work also use cluster controids but with a totally different way, which we will introduce in detail.

Under specific conditions, there are other depth extract methods, such as dark channel prior proposed by He *et al.* [23], that proved effective for hazed images.

The method closest to ours is the parametric model developed by Wang *et al.* [24] for describing the correlation between single color images and depth maps. This work treats the color image as a set of patches and derives the correlation with a kernel function in a nonlinear mapping space. They get convincing depth map through patch sampling. However, this work only demonstrates the effectiveness of the model and cannot estimate depth with an arbitrary input image. Our improvements are twofold: we extend this model from one image to many, and we transfer parameter set to an arbitrary input image according to the best image set match.

B. Parameter Transfer

We carry out a survey on transfer methods in the field of depth estimation. The nonparametric scene parsing by Liu *et al.* [25] avoids explicitly defining a parametric model and scales better with respect to the training data size. The depth transfer method by Karsch *et al.* [13] leverages this work and assumes that scenes with similar semantics should have similar depth distributions after densely aligned. Their method contains three stages. First, given an input image, they find K best matched images in RGB space. Then, the K images are

warped to be densely aligned with the input. Finally, they use an optimization scheme to interpolate and smooth the warped depth values to get the depth of the input.

Our work is different in three aspects. First, instead of depth, we transfer parameter set to the input image, so we do not need post process like warping. Second, our database is composed of parameter sets instead of RGB-D images, so the database occupies little space. Third, the depth values are computed with the transferred parameter set directly, so we do not need an optimization procedure after transfer.

III. Depth Estimation by Parameter Transfer

In this section, we first introduce the modeling procedure for inferring the correlation between color images and depth maps. Then, we introduce the parameter transfer method in detail.

A. Parametric Model

The prior work of Wang *et al.* [24] proposed a model to build the correlation between a single image I and its corresponding depth map D with a set of parameters. We extend this using a set of similar images IS and their corresponding depth map DS. Therefore, the parameters contain information of all the images in the set.

We regard to each color image as a set of overlapped fixed-size color patches, of which the size will be discussed later. For each image, we sample the patches $x_1, x_2, \ldots, x_p$ and their corresponding depth values from RGB-D image set. To avoid overfitting, we sample only p patches from each image. In our experiment, we set p as 1000 and the samples account for 0.026% of the total patches in one image. We use a uniform sampling method, i.e., we separate the image into grids and select samples uniformly from all the grids. By denoting N the number of images in an image set, totally we sample $N \times p$ patches. Specially, for a single image, $N = 1$.

1) Modeling the Correlation Between Image and Depth: After the sampling procedure, we model the correlation by measuring the sum squared error between the depth $\hat{\mathbf{d}}$ mapped with the sampled color patches and the ground-truth depth $\mathbf{d}$. The model is written as

$$E = \sum_{i=1}^{p \times N} \left| \text{tr} \left(W^T \sum_{j=1}^{n} \gamma_j \phi(x_i * f_j) \right) - d_i \right|^2 \tag{1}$$

where E is the sum squared estimation error, p is the number of sample patches per image, N is the number of images in the image set, f_j is the filters, and n is the number of filters and set as 9 in all the experiments. If set larger, the algorithm is expected to get better results but at a larger cost. ϕ is the kernel function to map the convolved patches and sum them up to *one patch*, γ_j is the weight of each convolved patch, and W is the weight matrix, whose size is the same as the size of the *one patch*, aiming at integrating the overall information from each patch.

Equation (1) can be rewritten as

$$E = \sum_{i=1}^{p \times N} |\mathbf{w}^T \phi(X_i F)\gamma - d_i|^2 \tag{2}$$

where X_i is a matrix reshaped from patch x_i. The row size of X_i is the same as that of f_i, while $F = [f_1, f_2, \ldots, f_n]$, $\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_n]^T$. $\mathbf{w}$ is the result of concatenating all the entries of W.

At the image level, F describes the texture gradient cues of the RGB image by extracting the frequency information. γ describes the variance of filters. We use principle component analysis (PCA) to initialize F and optimize it afterward. As for the size of filter, we need to balance between efficiency and effect. However, we use W to integrate the global information, so we can choose smaller sized filters to reduce time consumption. $\phi(\cdot)$ is set as $\phi(x) = \log(1 + x^2)$, as it has been proven effective in [24].

2) Estimating Model Parameters: First, we rewrite (2) as

$$E = \|M\phi(XF)\gamma - \mathbf{d}\|_2^2 \tag{3}$$

and

$$E = \|\Gamma\phi(F^T \hat{X})\mathbf{w} - \mathbf{d}\|_2^2 \tag{4}$$

where X is got by concatenating all the X_i in (2). $\hat{X}$ is got by concatenating all the X_i^T. Each row of M is $\mathbf{w}^T$ and each row of Γ is γ^T. Hence, (3) is a least square problem of γ and (4) is a least square problem of $\mathbf{w}$. Then we minimize E by optimizing the filters F. Finally, we get a set of parameters, consisting of F, γ, and $\mathbf{w}$. The detailed method for solving this can be found in [24].

B. Parameter Transfer

Our parameter transfer procedure outlined in Fig. 2 has three stages. First, we build a parameter set database using training RGB-D images. Second, given an input image, we find the most similar image sets using high-level image features and transfer the parameter set to the input image. Third, we compute the depth of the input image.

1) Parameter Set Database Building: Given a RGB-D training data set, we compute high-level image features for each image. Here, we use GIST [26] features, which can be used to measure similarities of images. Then, we categorize the training images to N sets, using the k-means cluster method. Next, we get the central GIST feature for each image set. For each image set, the corresponding parameter set is obtained using our parameter estimate model. The central GIST features and corresponding parameter sets comprise our parameter set database. Actually, this database is so small as to occupy much less space compared with the RGB-D data sets.

2) Image Set Matching: Given an input image, we compute its GIST feature and find the best matched central GIST feature from our trained database. Then the parameter set corresponding to the best matched central GIST feature (i.e., the central GIST feature of the most similar image set) is transferred to the input image. We define the best match as

$$G_{\text{best}} = \min_{i=1,2,\ldots,N} \|G_{\text{input}} - G_i\| \tag{5}$$

where G_{input} denotes the GIST feature of the input image and G_i denotes the central GIST feature of each image set.

As the most similar image set matches the input closely in feature space, the overall semantics of the scenes are similar.

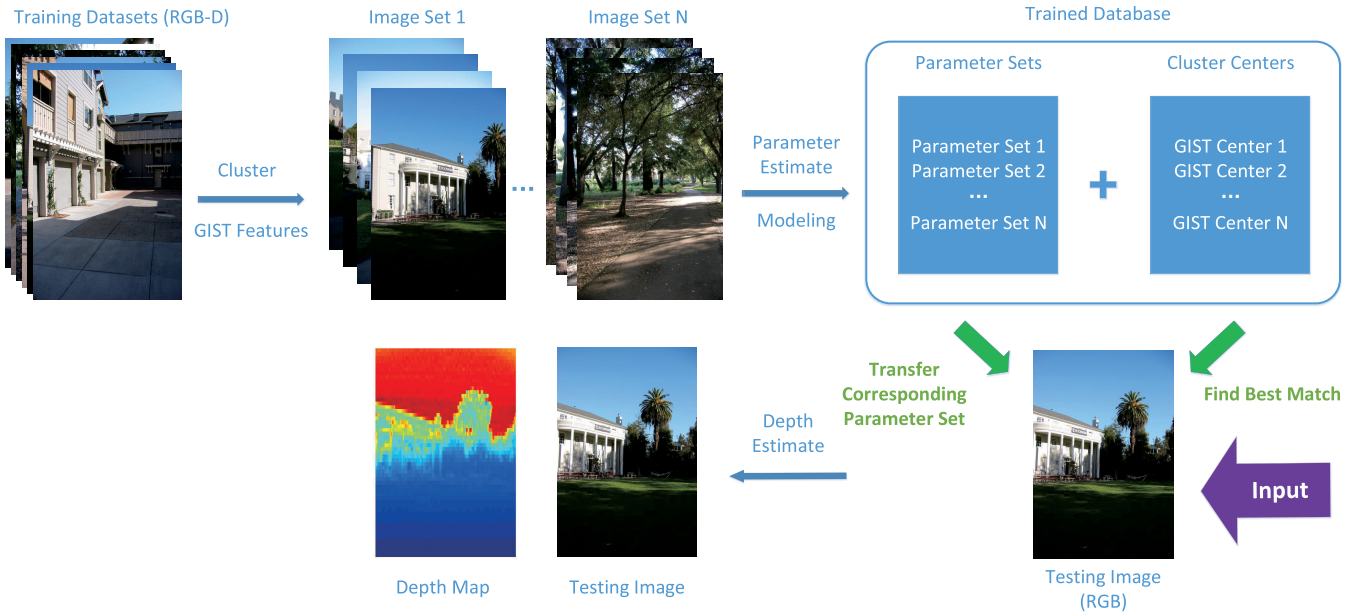


Fig. 2. Our pipeline for estimating depth. First, we build a parameter set database and then the parameter set is transferred to the input image according to the best matched GIST feature. Finally, the parameter set is used to estimate the depth.

At the low level, the cues, such as the texture gradient, texture variation, and color, are expected to be roughly similar to some extent. With the model above, the parameters connecting the images and depth maps should be similar. Therefore, it is reasonable to transfer the parameter set to the input image.

3) Depth Estimation: We use the color patches of the input image and the transferred parameter set to map the estimation depth. The computational formula is

$$\hat{\mathbf{d}} = M\phi(XF)\gamma \tag{6}$$

where X are the patches, F are the filters, γ is the weight to balance the filters, and M is the weight matrix. These parameters are all from the parameter set.

IV. Experiment

In this section, we evaluate the effectiveness of our DEPT method on single image RGB-D data sets.

A. RGB-D Data Sets

We use the Make3D range image data set [27]. The data set is collected using a 3D scanner and the corresponding depth maps using lasers. There are 534 images separated into two parts, of which one is the training part containing 400 images and the other is the testing part containing 134 images, respectively. The color image resolution is 2272×1704 and the ground-truth depth map resolution is 55×305. Before training, we resize the depth map resolution to the same size of the color image, so RGB and D (depth) have pixelwise correspondence.

B. Image Cluster

We compute the GIST features for each image in the training data set. Then we use the k-means algorithm to cluster the images into N sets, where we set N as 30. The images are well separated according to the scene semantics. The silhouette plot in Fig. 3 measures how well separated the resulting image sets are. Lines on the right side of 0 measure how distant that image is from neighboring image sets. Lines on the left of 0 indicate that the image is probably assigned to the wrong set. The vertical axis indicates different clusters (image sets). As we can see, most of the images are well clustered. As for the choosing of N, initially we choose it by observing the silhouette plot and then we try a series of values with a step of 10. The results around 30 are close, and 30 is the best. The cluster number can also be set according to existing pattern classification methods (e.g., methods to find best k in k-means algorithm [28], [29]). We believe N should not be too large or too small. Too large N may set similar scenes apart, while too small N may result in large scene variety in one set.

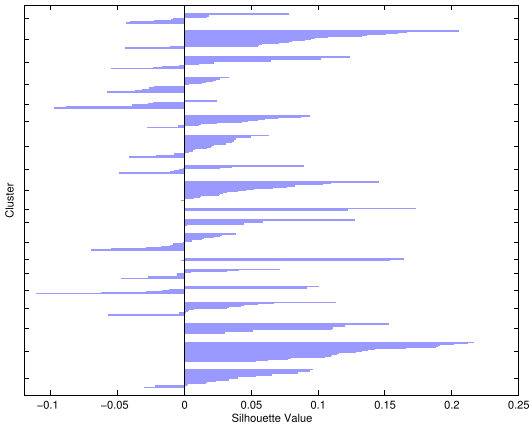
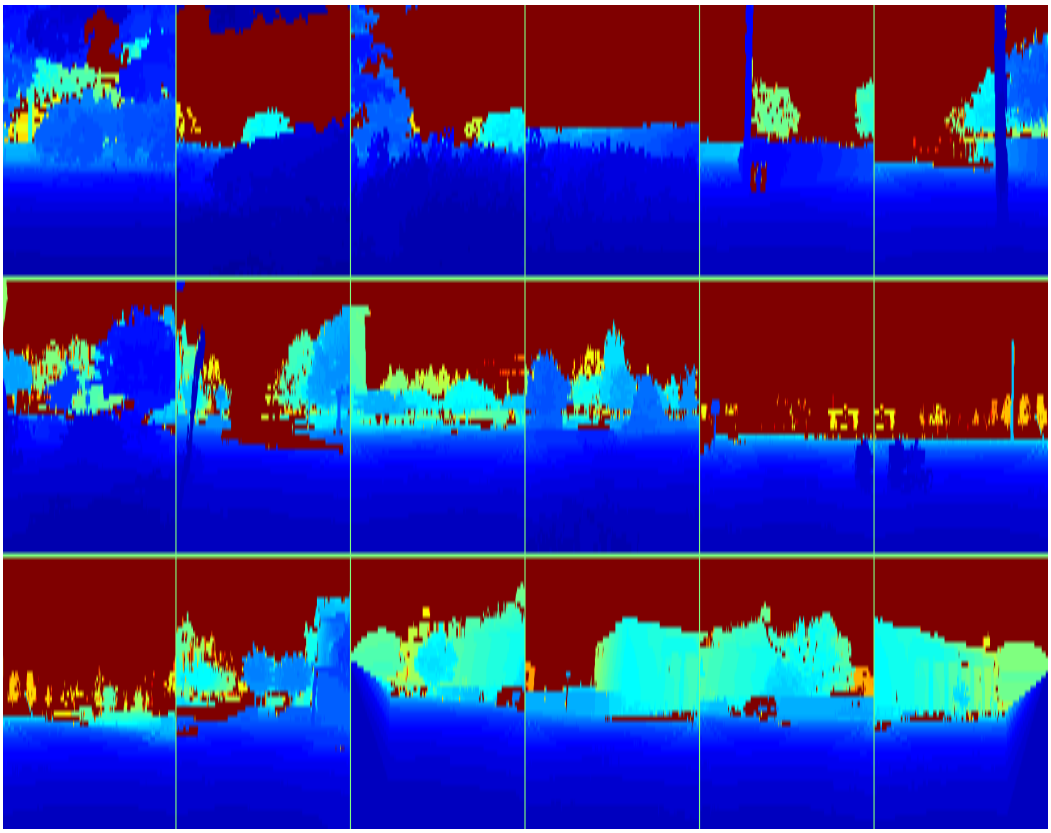


Fig. 3. Silhouette plot of the k-means cluster result. Each line represents an image. Lines on the right side of 0 measure how distant that image is from neighboring image sets. Lines on the left of 0 indicate that the image is probably assigned to the wrong set. The vertical axis indicates different clusters (image sets).

An example image set is shown in Fig. 4. It can be seen that the clustered images have roughly a similar semantic scene. The depth distributions also seem similar, as are shown in the color images as well as the depth maps.

(a)



(b)

Fig. 4. One example image set after the image cluster procedure. (a) Clustered image set containing 18 semantic similar images and (b) their corresponding depth maps. The depth distributions in the images are roughly similar.

C. Parameter Set Estimation

For each image set, we estimate the corresponding model parameters. The overlapped patch size is set 15×15. The filter size is set as 3×3. We separate each image into grids and uniformly sample 1000 patches per image. Therefore, for an N sized image set, totally $1000 \times N$ patches are sampled,

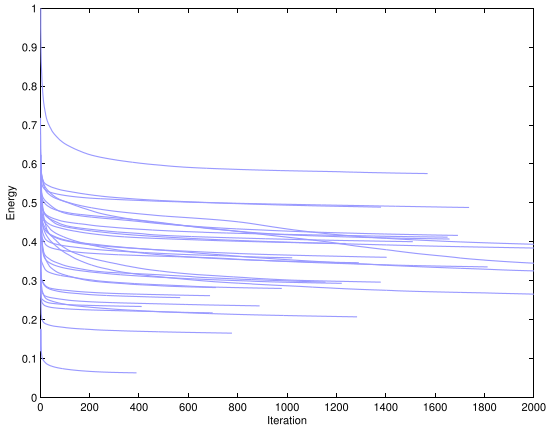


Fig. 5. Energy decline curves of the 30 image sets. E is on a ln scale.

which occupy 0.026% of the whole image set. We initialize the filters with the PCA method and optimize all the parameters using a warm-start gradient descent method. The iteration stop condition is $E < 10^{-6}$. In our experiment, the energy (i.e., the sum squared errors E) declines as Fig. 5 shows. As can be seen, most of the curves come to a steady state after about 1000 iterations. The smaller the steady energy is, the more similar the images in that set are.

For each image set, we obtain one optimized parameter set. The 30 parameter sets and the corresponding cluster centroids (the center of the GIST features in each image set) make up the parameter sets database.

D. Depth Estimation by Parameter Transfer

For each of the testing 134 images, we find the best matched image set from the parameter set database and compute the depth maps using the computational formula of (6).

1) Quantitative Comparison With Previous Methods: We calculate three common error metrics for the estimated depth. Denoting $\hat{\mathbf{D}}$ the estimated depth and $\mathbf{D}$ the ground-truth depth, we calculate *relative error*

$$\mathbf{RE} = \frac{|\hat{\mathbf{D}} - \mathbf{D}|}{\mathbf{D}} \tag{7}$$

LE ($\log_{10}$ *error*)

$$\mathbf{LE} = |\log_{10}(\hat{\mathbf{D}}) - \log_{10}(\mathbf{D})| \tag{8}$$

and *root-mean-squared error*

$$\mathbf{RMSE} = \sqrt{\sum_{i=1}^{P} (\hat{\mathbf{D}}_i - \mathbf{D}_i)^2 / P} \tag{9}$$

where P is the pixel number of a depth map.

Error measure for each image is the average value of all the pixels on the ground-truth resolution scale (55×305). Then the measures are averaged over all the 134 images to get final error metrics, which are listed in Table I.

As can be seen, our results are better than Depth MRF [9] in view of **RE** and **LE** and better than Make 3D [27] in view of **LE**. In total, the results of DEPT are comparable to those of the state-of-the-art learning-based automatic methods. Especially, DEPT requires only a very small-sized database, and once the database is built, we can compute the depth directly. Built from the 400 training RGB-D images that occupy 628 MB space, our database size is only 188 kB (0.03%). In contrast, the trained database of depth transfer [13] occupies 2.44 GB[1] (about four times that of the original data set size). Though our method has a *disadvantage* in average errors over the depth transfer, we have a huge *advantage* in database space consumption and computer performance requirement [Karsch *et al.* [13] claim depth transfer requires a great deal of data (GB scale) to be stored concurrently in memory in the optimization procedure], which are especially crucial when the database grows in real applications. Recent deep convolutional neural network (CNN)-based depth estimation methods get lower errors. Essentially, our convolutional operation and optimization method is similar to CNN with only one layer. From this point of view, our method achieves comparable results with fewer parameters. If implemented on a high-end GPU, our method would achieve much higher efficiency.

TABLE I

AVERAGE ERROR AND DATABASE SIZE COMPARISON OF VARIOUS ESTIMATE METHODS

Method	RE	LE	RMSE	Trained Database
Depth MRF [9]	0.530	0.198	16.7	-
Make3D [27]	0.370	0.187	-	-
Feedback Cascades [12]	-	-	15.2	-
Deep CNN Fields [20]	0.314	0.119	8.60	140 MB
Depth Transfer [13]	0.361	0.148	15.1	2.44 GB
DEPT with GIST(ours)	0.489	0.182	16.9	1.47 MB
DEPT with CNN(ours)	0.421	0.172	16.7	1.25 MB

Furthermore, our method has also advantages in some of the estimation effects, as is detailed in the following qualitative evaluation.

2) Qualitative Evaluation: A qualitative comparison of our estimated depth maps, depth maps estimated by depth transfer [13], and the ground-truth depth maps is demonstrated in Figs. 6 and 7. As can be seen, our estimated depth maps are visually reasonable and convincing, especially in the details like texture variations (e.g., the tree in the second column of Fig. 6) and relative depth (e.g., the pillars' depth in the last column of Fig. 6 is well estimated by our DEPT method, whereas depth transfer [13] estimates wrong). Actually, some of our results are even more accurate than the ground truth (e.g., in the third column in Fig. 7, there is a large part of a wrong depth in the building area of the ground-truth depth map). The ground-truth maps have some scattered noises, which may result from the capturing device, while the noises in our depth maps are fewer because we use the overall information in the image set. However, we must point out that the sky areas in our depth maps are not as pleasing, which may result from the variation of sky color and texture among various images in a set, especially when the cluster result is

[1]Implemented with the authors' public codes at http://research.microsoft.com/en-us/downloads/29d28301-1079-4435-9810-74709376bce1/.

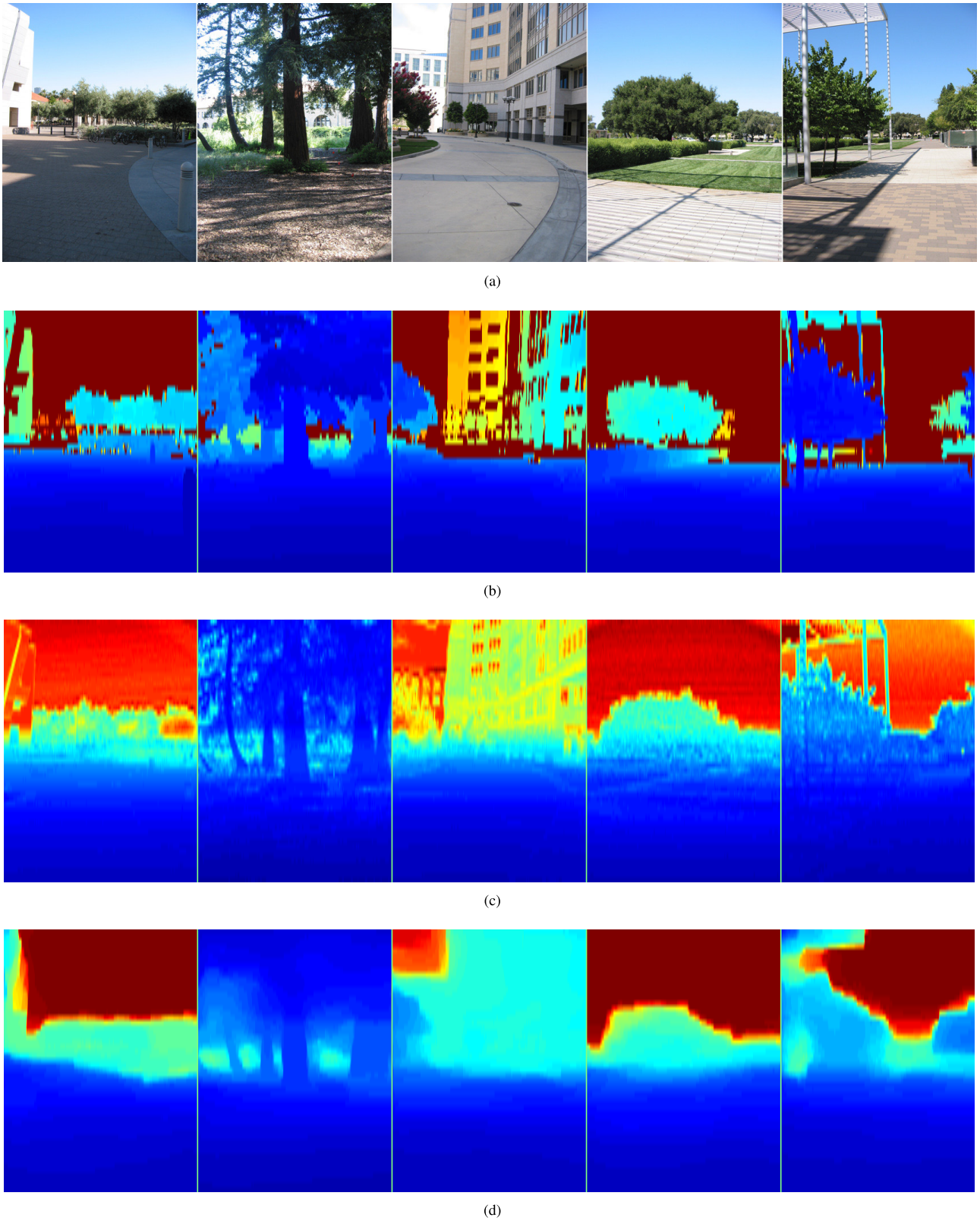
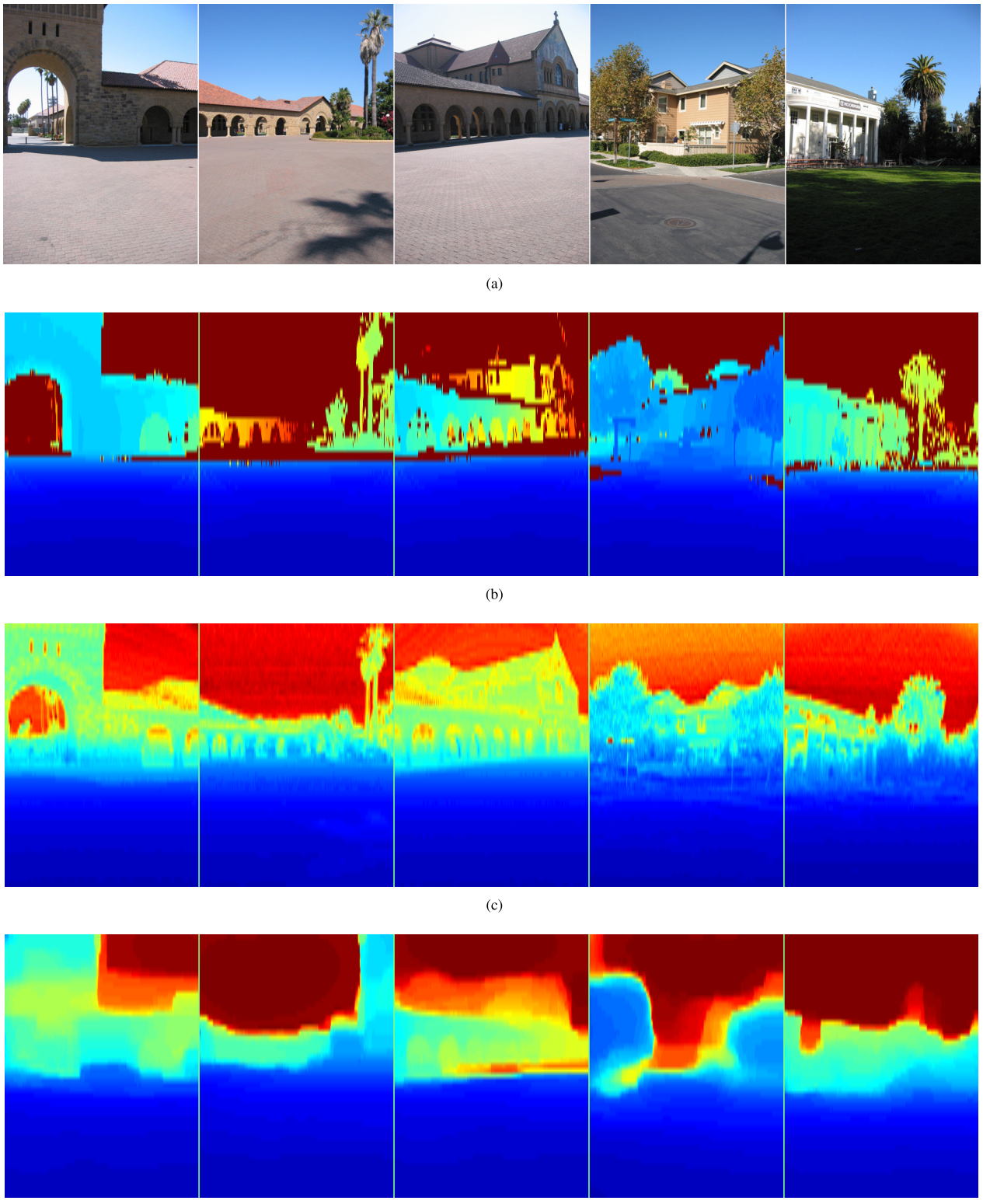


Fig. 6. Performance comparison: scenes of streets, squares, and trees. (a) Some test images containing streets, squares, or trees. (b) Corresponding ground-truth depth maps. (c) Estimated depth maps by DEPT (our method). (d) Estimated depth maps by depth transfer [13].

biased. This may result in the increase in average error in the previous metrics. However, as the RGB-D images acquired by depth imaging devices increase, our database can expand easily due to the extremely small space consumption, which means that we may get more and more accurately matched parameter sets for existing RGB images and video frames.



(a) (b) (c) (d)

Fig. 7. Performance comparison: scenes of buildings. (a) Some test images containing buildings. (b) Corresponding ground-truth depth maps. (c) Estimated depth maps by DEPT (our method). (d) Estimated depth maps by depth transfer [13].

E. Evaluation on Indoor Data Sets

We also implement an experiment on the NYU Depth V2 data set [30], which consists of 1449 indoor RGB-D images captured with Kinect. We use the *labeled data set*,[2] i.e., 1449 densely labeled pairs of aligned RGB and depth images. The data set is partitioned into 795 training images and 654 testing images. When training DEPT, we cluster the training data set to 80 sets, guided by a k-means silhouette plot and linear search. One example of the cluster set is shown in Fig. 8. The quantitative results are

[2]http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

Fig. 8. One example image set after the image cluster procedure on NYU. The clustered image set contains nine semantic similar images.

shown in Table II. In addition to the three standard metrics, we also report the metrics used in [14], defined as

$$\frac{1}{N}\sum_{p=1}^{N}\left[\left[\max\left(\frac{d_p}{g_p},\frac{g_p}{d_p}\right)=\delta<t\right]\right]\times 100\% \tag{10}$$

where g_p is the ground truth of pixel p, d_p is the corresponding estimated depth, N is the number of pixels, $t = 1.25, 1.25^2, 1.25^3$ is the threshold, and $[[\cdot]]$ denotes the indicator function. We can observe that DEPT achieves comparable quantitative results with much less space and time consumption.

The qualitative results are shown in Fig. 9. We can see that DEPT gets not so smooth results (we did not use the smoothing operation as did depth transfer), but infers more details on the edges. This may be useful when an application cares more about edges of the depth map.

In addition, the testing procedure (654 images) consumes about 4 h with DEPT on our computer (Intel Xeon E3-1330 V2 CPU, 16-GB RAM, 64-b Windows 7, without any algorithm optimization), while it takes about 45 h with depth transfer [13].

F. Replace GIST With Deep CNN

Following [31] and [32], we observe that CNNs have good scene descriptions for images. Thus, we follow the method of [32] to compute the representations of the RGB images. The CNN feature extraction process is illustrated in Fig. 10. For each of the training images, the representation is computed as follows:

$$\upsilon = \mathrm{CNN}_{\theta_c}(I) \tag{11}$$

where $\mathrm{CNN}_{\theta_c}(I)$ transforms the pixels of image I into a 4096D activation of the fully connected layer immediately before the classifier, i.e., the 1000-way softmax layer. The CNN parameters θ_c contain approximately 60 million parameters and the architecture closely follows the network of Krizhevsky *et al.* [33], but we chop off the final 1000-way softmax layer. In this way, after network forwarding, each image is represented as a 4096D vector.

This vector can be treated as CNN features of the image. We replace GIST features with CNN features in the previous framework.

We carry out experiments with the new framework. The result is listed in Tables I and II. We observe a decrease in all the error indicators. This performance is better than the originally proposed method in the conference version of this work [34]. In the meantime, though CNN features can improve the performance, it increases time consumption and model size, because for now, most of the CNN-based methods rely on high-performance GPUs. It is too slow on personal computers, not to mention mobile devices. Therefore, we need

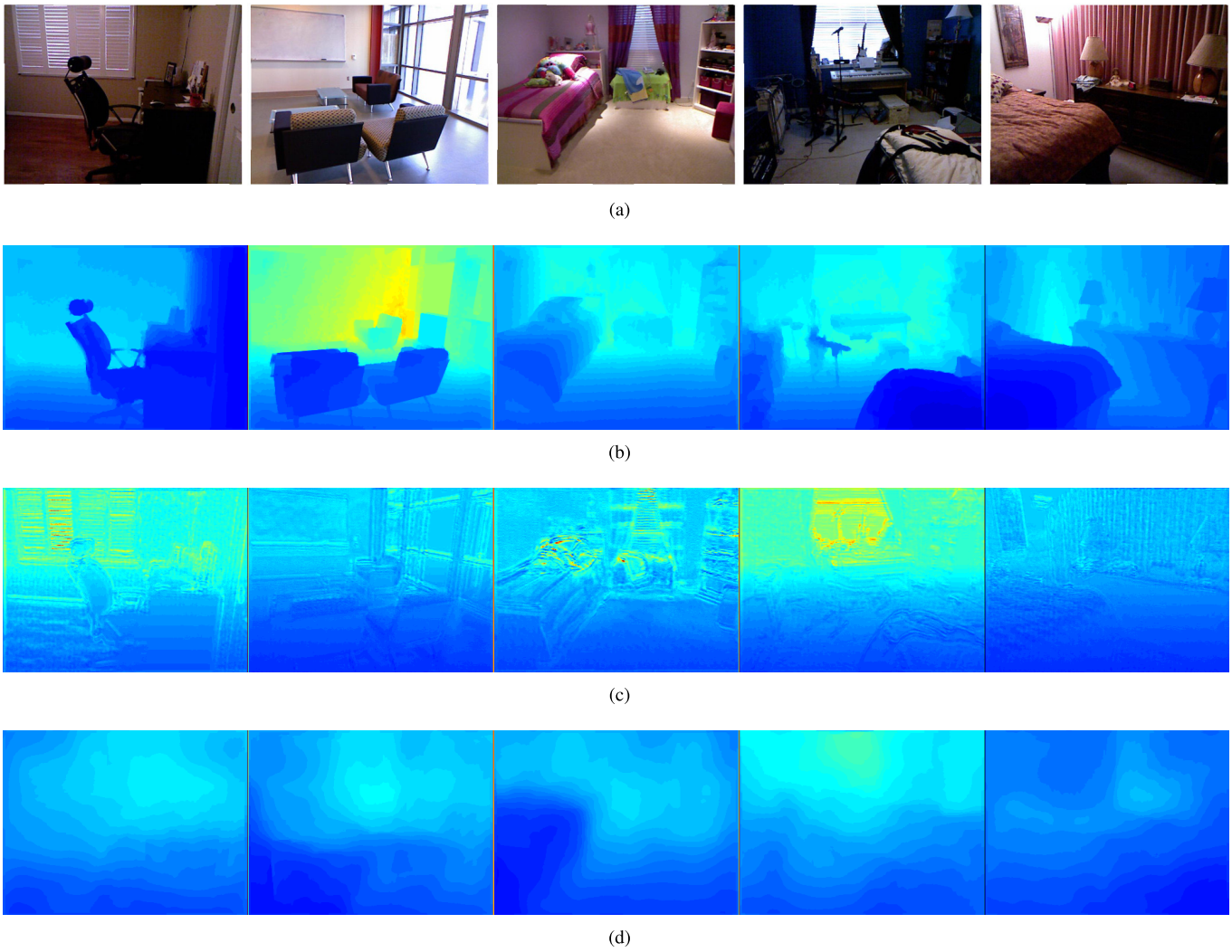


Fig. 9. Performance comparison: indoor scenes. (a) Some test images containing indoor scenes. (b) Corresponding ground-truth depth maps. (c) Estimated depth maps by DEPT (our method). (d) Estimated depth maps by depth transfer [13].

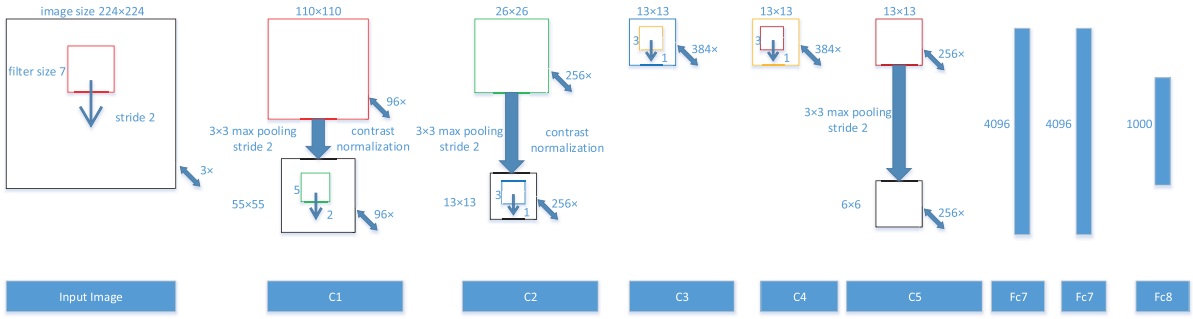


Fig. 10. Illustration of the CNN feature extraction architecture. A 224×224 crop of an image (RGB) is presented as the input. It is convolved with 96 different filters, each of size 7×7, using a stride of 2 in both x and y. The resulting feature maps are then passed through a rectified linear function (not shown), pooled (max within 3×3 regions, using stride 2), and contrast normalized across feature maps to give 96 different 55×55 element feature maps. Similar operations are repeated in layers 2–5. The last two layers are fully connected, taking features from the top convolutional layer as input in vector form ($6 \times 6 \times 256 = 9216$ dimensions). The output of layer 7 are our CNN features in vector form (4096 dimensions). The final layer is a 1000-way softmax function, whose output is one predicted class out of 1000.

to balance the performance, speed, and model size in real applications. However, we can expect more improvement of DEPT when better algorithms for semantic scene matching are proposed.

TABLE II
EXPERIMENTAL RESULTS ON THE NYU INDOOR DATA SET

Method	RE	LE	RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Database	Time
Depth Transfer [13]	0.374	0.134	1.12	49.81%	79.46%	93.75%	1.14 GB	45 hours
DC-Depth [18]	0.335	0.127	1.06	51.55%	82.32%	95.00%	-	-
Zhuo *et. al* [19]	0.305	0.122	1.04	52.50%	83.77%	96.16%	-	-
DEPT with GIST (ours)	0.392	0.151	1.19	48.50%	79.02%	93.12%	465 KB	4 hours
DEPT with CNN (ours)	0.353	0.130	1.11	51.24%	80.62%	94.35%	460 KB	4 hours

V. CONCLUSION

In this paper, we propose a lightweight, effective, and fully automatic technique to restore depth information from single still images. Our DEPT method is novel in that we use clustered scene semantics similar image sets to model the correlation between RGB information and D (depth) information, obtaining a database of parameter sets and cluster centers. DEPT requires only the trained parameter set database, which occupies much less space compared with previous learning-based methods. Experiments on RGB-D benchmark data sets show quantitatively and qualitatively good results comparable to the state-of-the-art results. The estimated depth maps are visually reasonable and convincing, especially in the details, such as texture variations and relative depth. Furthermore, as RGB-D images acquired by depth imaging devices increase, our database can expand easily due to the extremely small space consumption. As our model is only about 1 MB, it is very suitable to use on mobile devices (the code will be released upon publication). In the future work, we would like to improve the cluster accuracy by exploring more accurate similarity metrics that are applicable to our image and depth correlation model. We plan to build a larger RGB-D image data set, as more data bring better performance with our method. Finally, we suppose it is also meaningful to improve the depth estimation performance for video frames using optical flow features or other features related to time coherence.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for helpful comments.

REFERENCES

[1] Q. Liu, Y. Yang, R. Ji, Y. Gao, and L. Yu, "Cross-view down/up-sampling method for multiview depth video coding," *IEEE Signal Process. Lett.*, vol. 19, no. 5, pp. 295–298, May 2012.

[2] Y. Liu, Q. Dai, and W. Xu, "A point-cloud-based multiview stereo algorithm for free-viewpoint video," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 3, pp. 407–418, May/Jun. 2010.

[3] F. Li, J. Yu, and J. Chai, "A hybrid camera for motion deblurring and depth map super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[4] A. Torralba and A. Oliva, "Depth estimation from image structure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1226–1238, Sep. 2002.

[5] Y. Horry, K.-I. Anjyo, and K. Arai, "Tour into the picture: Using a spidery mesh interface to make animation from a single image," in *Proc. 24th Annu. Conf. Comput. Graph. Interact. Techn.*, 1997, pp. 225–232.

[6] L. Zhang, G. Dugas-Phocion, J.-S. Samson, and S. M. Seitz, "Single-view modelling of free-form scenes," *J. Visualizat. Comput. Animation*, vol. 13, no. 4, pp. 225–235, 2002.

[7] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 577–584, 2005.

[8] E. Delage, H. Lee, and A. Y. Ng, "A dynamic Bayesian network model for autonomous 3D reconstruction from a single indoor image," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2006, pp. 2418–2428.

[9] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1161–1168.

[10] A. Saxena, S. H. Chung, and A. Y. Ng, "3-D depth reconstruction from a single still image," *Int. J. Comput. Vis.*, vol. 76, no. 1, pp. 53–69, 2008.

[11] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1253–1260.

[12] C. Li, A. Kowdle, A. Saxena, and T. Chen, "Towards holistic scene understanding: Feedback enabled cascaded classification models," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2010, pp. 1351–1359.

[13] K. Karsch, C. Liu, and S. B. Kang, "Depth transfer: Depth extraction from video using non-parametric sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2144–2158, Nov. 2014.

[14] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 89–96.

[15] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2014, pp. 2366–2374.

[16] D. Eigen and R. Fergus. (Dec. 2015). "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture." [Online]. Available: http://arxiv.org/abs/1411.4734

[17] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2800–2809.

[18] M. Liu, M. Salzmann, and X. He, "Discrete-continuous depth estimation from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 716–723.

[19] W. Zhuo, M. Salzmann, X. He, and M. Liu, "Indoor scene structure analysis for single image depth estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 614–622.

[20] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5162–5170.

[21] M. H. Baig, V. Jagadeesh, R. Piramuthu, A. Bhardwaj, W. Di, and N. Sundaresan, "Im2depth: Scalable exemplar based depth transfer," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2014, pp. 145–152.

[22] M. H. Baig and L. Torresani. (2015). "Coarse-to-fine depth estimation from a single image via coupled regression and dictionary learning." [Online]. Available: http://arxiv.org/abs/1501.04537v2

[23] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.

[24] Y. Wang, R. Wang, and Q. Dai, "A parametric model for describing the correlation between single color images and depth maps," *IEEE Signal Process. Lett.*, vol. 21, no. 7, pp. 800–803, Jul. 2014.

[25] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2368–2382, Dec. 2011.

[26] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[27] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.

[28] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[29] G. Hamerly and C. Elkan, "Learning the k in k-means," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2004, vol. 16, p. 281.
[30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Computer Vision*. Berlin, Germany: Springer, 2012, pp. 746–760.
[31] K. Kang and X. Wang. (2014). "Fully convolutional neural networks for crowd segmentation." [Online]. Available: https://arxiv.org/abs/1411.4464
[32] A. Karpathy and L. Fei-Fei. (2014). "Deep visual-semantic alignments for generating image descriptions." [Online]. Available: https://arxiv.org/abs/1412.2306
[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, pp. 1097–1105.
[34] X. Li, H. Qin, Y. Wang, Y. Zhang, and Q. Dai, "Dept: Depth estimation by parameter transfer for single still images," in *Proc. 12th Asian Conf. Comput. Vis. (ACCV)*, Nov. 2014, pp. 45–58.

Hongwei Qin (S'16) received the B.S. degree from Tsinghua University, Beijing, China, in 2012, where he is currently pursuing the Ph.D. degree with the Department of Automation.

His current research interests include deep learning for visual recognition, 3D reconstruction, and object detection.

Xiu Li (M'15) received the Ph.D. degree in computer integrated manufacturing from the Nanjing University of Aeronautics and Astronautics in 2000.

She has been with Tsinghua University, Beijing, China. Her research interests include data mining, business intelligence systems, knowledge management systems, and decision support systems.

Yangang Wang received the B.E. degree from the School of Instrument Science and Engineering, Southeast University, Nanjing, China, in 2009, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2014, under the supervision of Prof. Q. Dai.

He is currently an Associate Researcher with Microsoft Research Asia, Beijing. His research interests include computer vision, computer graphics, and computational photography.

Yongbing Zhang (M'15) received the B.A. degree in English and the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2004, 2006, and 2010, respectively.

He is currently with the Graduate School at Shenzhen, Tsinghua University, Shenzhen, China. His research interests include video processing, image and video coding, video streaming, and transmission.

Qionghai Dai (SM'05) received the B.S. degree from Shanxi Normal University, Xi'an, China, in 1987, and the M.E. and Ph.D. degrees from Northeastern University, Shenyang, China, in 1994 and 1996, respectively.

Since 1997, he has been with Tsinghua University, Beijing, China, where he is currently a Professor and the Director of the Broadband Networks and Digital Media Laboratory. His research interests include video communication, computer vision, and computational photography.