DEPT: Depth Estimation by Parameter Transfer for Single Still Images

Xiu Li[1,2], Hongwei Qin[1,2(✉)], Yangang Wang[3], Yongbing Zhang[1,2], and Qionghai Dai[1]

[1] Department of Automation, Tsinghua University, Beijing, China
{li.xiu,zhang.yongbing}@sz.tsinghua.edu.cn,
qionghaidai@tsinghua.edu.cn
[2] Graduate School at Shenzhen, Tsinghua University, Beijing, China
qhw12@mails.tsinghua.edu.cn
[3] Microsoft Research Asia, Beijing, China
yangangw@microsoft.com

Abstract. In this paper, we propose a new 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 compute 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 to the past learning-based methods, our trained database only consists of trained features and parameter sets, which occupy little space. We evaluate our depth estimation method on the benchmark RGB-D (RGB + depth) datasets. The experimental results are comparable to the state-of-the-art, demonstrating the promising performance of our proposed method.

1 Introduction

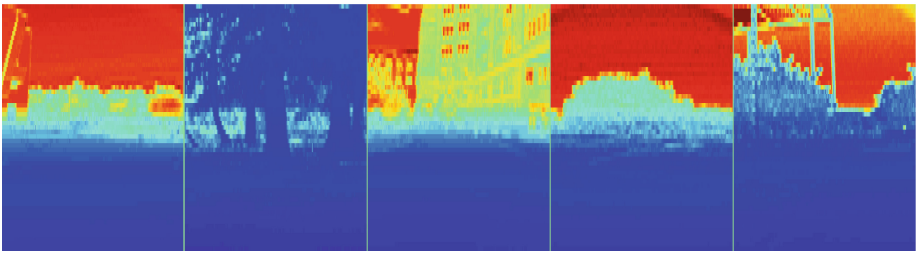
Images captured with conventional cameras lose the depth information of the scene. However, scene depth is of great importance for many computer vision tasks. 3D applications like 3D reconstruction for scenes (*e.g.*, Street View on Google Map), robot navigation, 3D videos, and free view video(FVV) [1] all rely on scene depth. Depth information can also be useful for 2D applications like image enhancing [2] and scene recognition [3]. Recent RGB-D imaging devices like Kinect are greatly limited on the perceptive range and depth resolution. Neither can they 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.

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-16808-1_4) contains supplementary material, which is available to authorized users.

D. Cremers et al. (Eds.): ACCV 2014, Part II, LNCS 9004, pp. 45–58, 2015.
DOI: 10.1007/978-3-319-16808-1_4

(a) Test images



(b) Estimated depth maps by DEPT

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.

We build the connection between image and depth with a set of parameters. A parameter sets 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, the paper is organized as follows. In Sect. 2, the related techniques are surveyed. In Sect. 3, we introduce our proposed DEPT (depth estimation by parameter transfer) method in details. We demonstrate our method on the RGB-D benchmark datasets in Sect. 4. Finally, we conclude our work in Sect. 5.

2 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.

2.1 Depth Estimation from Single Images

The reason 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, occlusions and so on. Global cues are also crucial to inferring depth, as

the ability humans have. So, integrating local and global cues of a single image to estimate depth is reasonable.

There are semi-automatic and automatic methods for depth estimation from single images. Horry *et al.* [4] propose *tour into the picture*, where the user interactively adds planes to an image to make animation. The work of Zhang *et al.* [5] requires the user to add constrains manually to images to estimate depth.

Automatic methods for single image depth estimation come up in recent years. Hoiem *et al.* [6] propose *automatic photo pop-up*, which reconstructs an outdoor image using assumed planar surfaces of it. Delage *et al.* [7] develop a Bayesian framework applied to indoor scenes. Saxena *et al.* [8] propose a supervised learning approach, using a discriminatively-trained Markov Random Field (MRF) that incorporates multi-scale local and global image features. Then, they improve this method in [9]. After that, depth estimation from predicted semantic labels is proposed by Liu *et al.* [10]. A more sophisticated model called Feedback Enabled Cascaded Classification Models (FE-CCM) is proposed by Li *et al.* [11]. One typical depth estimation method is Depth Transfer, developed by Karsch *et al.* [12]. This method first builds a large scale RGB-D images and features database, then acquires the depth of the input image by transferring the depth of several similar images after warping and optimizing procedures.

Under specific conditions, there are other depth extract methods, such as dark channel prior proposed by He *et al.* [13], proved effective for hazed images.

The method closest to ours is the parametric model developed by Wang *et al.* [14] 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 non-linear mapping space. They get convincing depth map through patch sampling. However, this work only demonstrates the effectiveness of the model, and can't estimate depth with an arbitrary input image. Our improvements are two-fold: we extend this model from one image to many, and we transfer parameter set to an arbitrary input image according to best image set match.

2.2 Parameter Transfer

We carry out a survey on transfer methods in the field of depth estimation. The non-parametric scene parsing by Liu *et al.* [15] avoids explicitly defining a parametric model and scales better with respect to the training data size. The Depth Transfer method by Karsch *et al.* [12] leverages this work and assumes that scenes with similar semantics should have similar depth distributions after densely aligned. Their method has 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 don't 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 don't need an optimization procedure after transfer.

3 DEPT: 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.

3.1 The Parametric Model

The prior work of Wang *et al.* [14] proposes 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 by using a set of similar images IS and their corresponding depth map DS. So the parameters contain information of all the images in the set.

We regard each color image as a set of overlapped fixed-size color patches. We will discuss the patch size later. For each image, we sample the patches $x_1, x_2, ..., x_p$ and their corresponding depth values from RGB-D image set. To avoid over-fitting, we only sample 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 as the number of images in an image set, totally we sample $N \times p$ patches. Specially, for single image, $N = 1$.

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

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

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, n is the number of filters, ϕ is the kernel function to map the convoluted patches and sum them up to *one patch*, γ_j is the weight of each convoluted patch, W is the weight matrix, whose size is the same 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 f_i, while $F = [f_1, f_2, ..., f_n]$, $\gamma = [\gamma_1, \gamma_2, ..., \gamma_n]^T$. $\mathbf{w}$ 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 afterwards. 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 consuming. $\phi(\cdot)$ is set as $\phi(x) = \log(1 + x^2)$, as it has been proven effective in [14].

Estimating Model Parameters. First, we rewrite Eq. 2 as

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

and

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

where X is got by concatenating all the X_i in Eq. 2. $\hat{X}$ is got by concatenating all the X_i^T. Each row of M is $\mathbf{w}^T$, and each row of $\mathbf{\Gamma}$ is γ^T. So Eq. 3 is a least square problem of γ, and Eq. 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 $\mathbf{w}$.

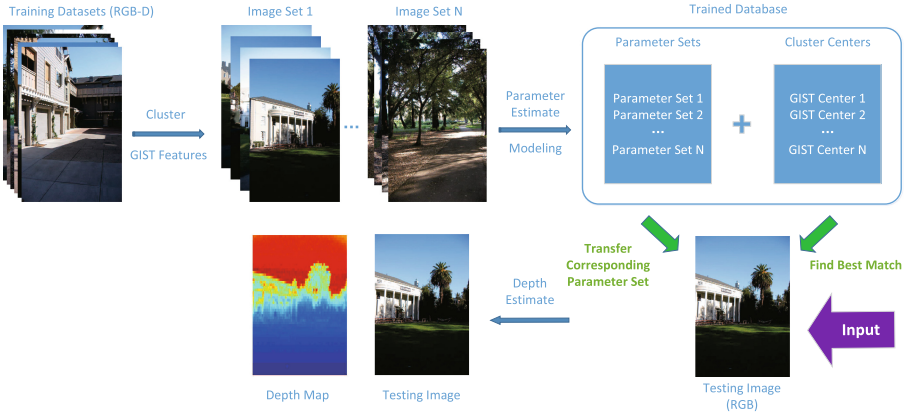


Fig. 2. Our pipeline for estimating depth. First we build a parameter set database, then the parameter set is transferred to the input image according to the best matched GIST feature. Finally, the parameter set is used to estimate the depth.

3.2 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.

Parameter Set Database Building. Given a RGB-D training dataset, we compute high-level image features for each image. Here, we use GIST [16] features, which can be used to measure similarities of images. Then, we category the training images to N sets, using KNN (K Nearest Neighbors) cluster method. And 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 compose our parameter set database. Actually, this database is so small as to occupy much less space compared to the RGB-D datasets.

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

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

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 match the input closely in feature space, the overall semantics of the scenes are similar. At the low level, the cues such as the texture gradient, texture variation, and color are expected to be roughly similar to some extend. With the model above, the parameters connecting the images and depth maps should be similar. So, it is reasonable to transfer the parameter set to the input image.

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

4 Experiment

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

4.1 RGB-D Datasets

We use the Make3D Range Image Dataset [17]. The dataset is collected using 3D scanner and the corresponding depth maps using lasers. There are totally 534 images separated into two parts, which are the training part containing 400 images and the testing part containing 134 images, respectively. The color image resolution is 2272×1704, and the ground truth depth map resolution is 55×305.

4.2 Image Cluster

We compute the GIST features for each image in the training dataset. Then we use KNN algorithm to cluster the images into N sets, here 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 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, we test 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 accurately set according to existing pattern classification methods (*e.g.* methods to find best k in k-means algorithm).

An example image set is shown in Fig. 5. It can be seen that the clustered images have roughly similar semantic scene. The depth distributions also seem similar, as are shown in the color images as well as the depth maps.

4.3 Parameter Sets 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. So for an N

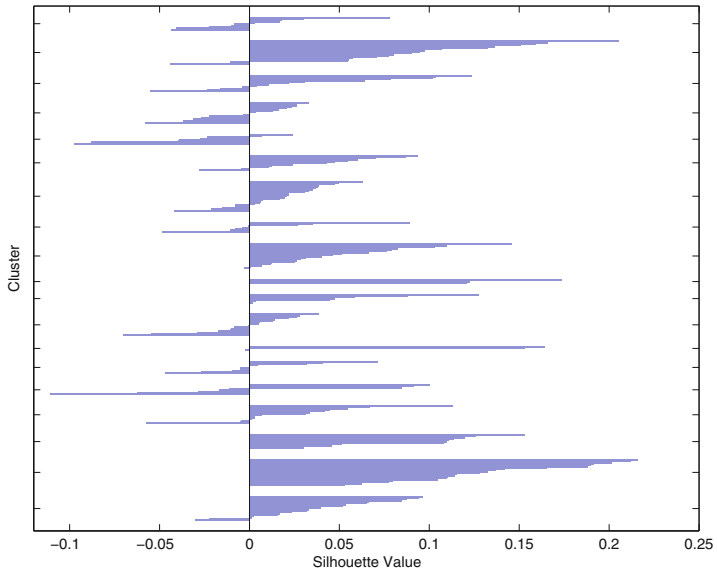


Fig. 3. Silhouette plot of the KNN cluster result. Each line represents an image. Lines on the right side of 0 measure how distant that image is from neighboring image sets. Lines on the left of 0 indicate that image is probably assigned to the wrong set. The vertical axis indicates different clusters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

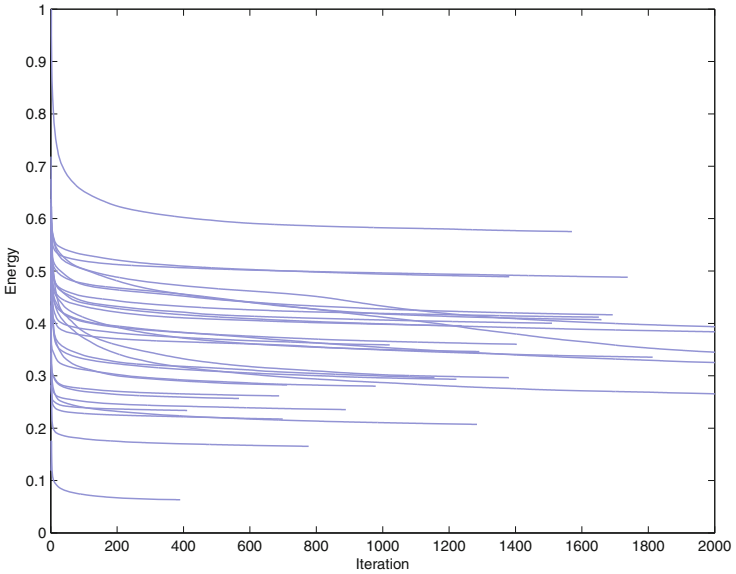


Fig. 4. Energy decline curves of the 30 image sets. E is on a ln scale.

sized image set, totally $1000 \times N$ patches are sampled, which occupy 0.026 % of the whole image set. We initialize the filters with PCA method, and optimize all the parameters using 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. 4 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 centers (the center of the GIST features in each image set) make up the parameter sets database.

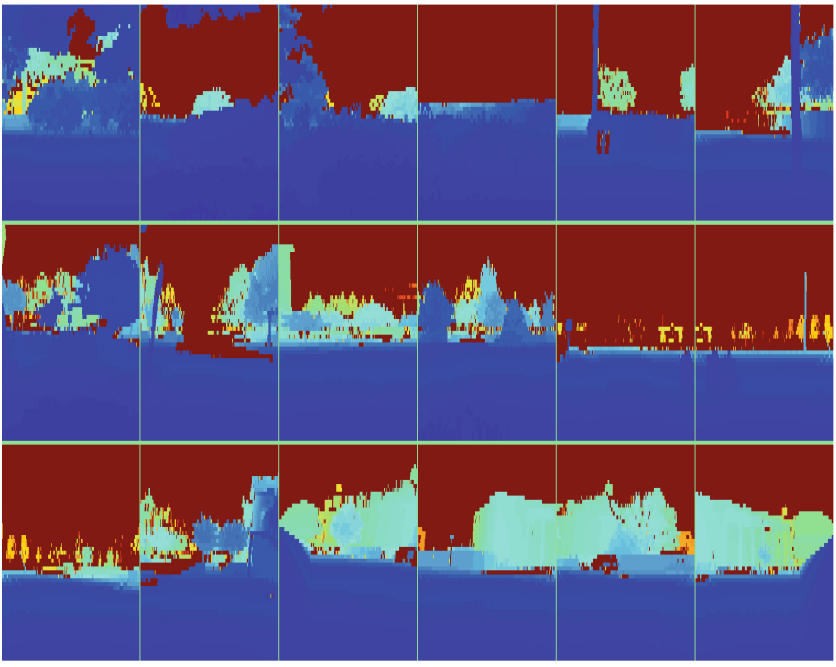
4.4 Depth Estimation by Parameter Transfer

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

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

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

(a) One clustered image set



(b) The corresponding depth maps

Fig. 5. One example image set after image cluster procedure. (a) is a clustered image set, containing 18 semantic similar images, (b) are their corresponding depth maps. The depth distributions in the images are roughly similar.

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

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

and **RMSE** (*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 1.

Table 1. Average error and database size comparison of various estimate methods.

Method	RE	LE	RMSE	Trained Database
Depth MRF [8]	0.530	0.198	16.7	-
Make3D [17]	0.370	0.187	-	-
Feedback Cascades [11]	-	-	15.2	-
Depth Transfer [12]	0.361	0.148	15.1	2.44 GB
DEPT(ours)	0.489	0.182	16.9	188 KB

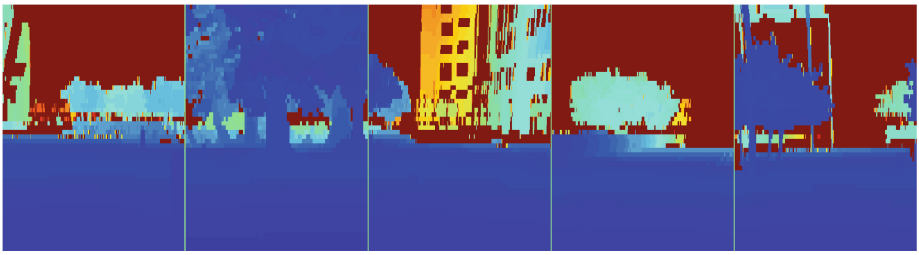
As can be seen, our results are better than Depth MRF [8] in view of **RE** and **LE**, better than Make3D [17] in view of **LE**. Totally speaking, the results of DEPT are comparable with the state-of-the-art learning based automatic methods. Especially, DEPT only requires a very small sized database, 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 %). As a contrast, the trained database of Depth Transfer [12] occupies 2.44 GB[1] (about 4 times of the original dataset size). Though our method has *disadvantage* in average errors over the Depth Transfer [12], we have large *advantages* in database space consuming and computer performance requirement(in [12], the authors 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.

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

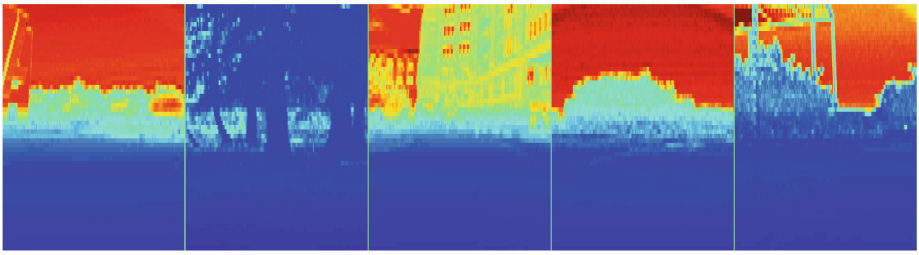
Qualitative Evaluation. A qualitative comparison of our estimated depth maps, depth maps estimated by Depth Transfer [12] and the ground truth depth

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

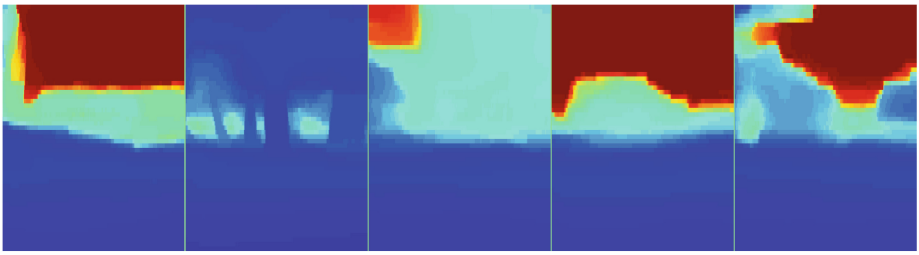
(a) Test images



(b) Ground truth depth maps



(c) Estimated depth maps by DEPT (our method)

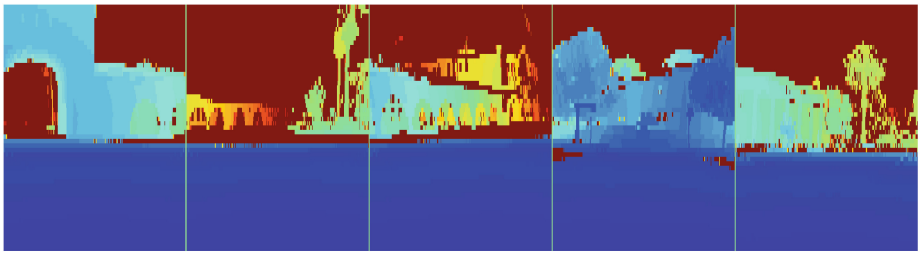


(d) Estimated depth maps by Depth Transfer [12]

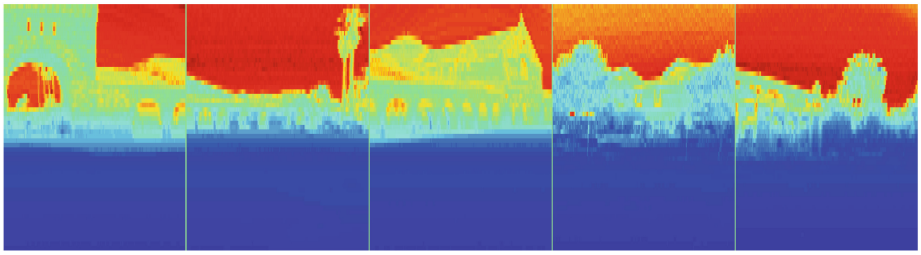
Fig. 6. Performance comparison: scenes of streets, squares and trees. (a) show some test images containing streets, squares or trees, (b) are corresponding ground truth depth maps, (c) are estimated depth maps by DEPT (our method), (d) are estimated depth maps by Depth Transfer [12]

maps are 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

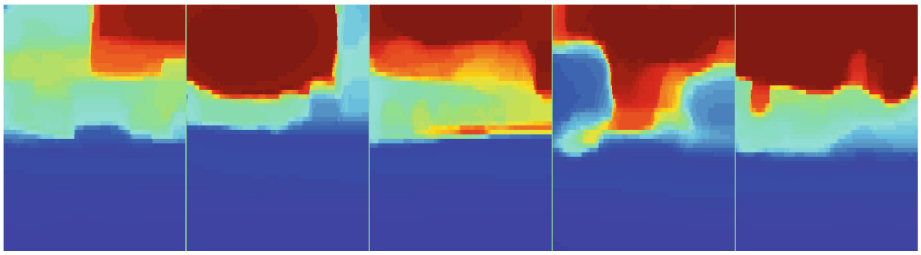
(a) Test images



(b) Ground truth depth maps



(c) Estimated depth maps by DEPT (our method)



(d) Estimated depth maps by Depth Transfer [12]

Fig. 7. Performance comparison: scenes of buildings. (a) show some test images containing buildings, (b) are corresponding ground truth depth maps, (c) are estimated depth maps by DEPT (our method), (d) are estimated depth maps by Depth Transfer [12]

(e.g., the pillars' depth in the last column of Fig. 6 is well estimated by our DEPT method, while Depth Transfer [12] 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 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 less because of the using of overall information in the image set. But 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 biased. This may result in the increase of average error in the previous metrics. However, as the increasing of RGB-D images acquired by depth imaging devices, our database can expand easily due to the extremely small space consuming, which means we may get more and more accurate matched parameter sets for existing RGB images and video frames.

5 Conclusion and Future Works

In this paper, we propose an effective and fully automatic technique to restore depth information from single still images. Our depth estimation by parameter transfer (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 only requires the trained parameter sets database which occupies much less space compared with previous learning based methods. Experiments on RGB-D benchmark datasets show quantitatively comparable to the state-of-the-art and qualitatively good results. The estimated depth maps are visually reasonable and convincing, especially in the details like texture variations and relative depth. Further more, as the increasing of RGB-D images acquired by depth imaging devices, our database can expand easily due to the extremely small space consuming. 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. And we suppose it is also meaningful to improve the depth estimation performance for video frames by using optical flow features or other features related to time coherence.

Acknowledgement. This work is supported by National Natural Science Foundation of China (Grant No. 71171121/61033005) and National 863 High Technology Research and Development Program of China (Grant No. 2012AA09A408).

References

1. Liu, Q., Yang, Y., Ji, R., Gao, Y., Yu, L.: Cross-view down/up-sampling method for multiview depth video coding. IEEE Sig. Process. Lett. **19**, 295–298 (2012)
2. Li, F., Yu, J., Chai, J.: A hybrid camera for motion deblurring and depth map super-resolution. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008, pp. 1–8. IEEE (2008)
3. Torralba, A., Oliva, A.: Depth estimation from image structure. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 1226–1238 (2002)

4. Horry, Y., Anjyo, K.I., Arai, K.: Tour into the picture: using a spidery mesh interface to make animation from a single image. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 225–232. ACM Press/Addison-Wesley Publishing Co. (1997)
5. Zhang, L., Dugas-Phocion, G., Samson, J.S., Seitz, S.M.: Single-view modelling of free-form scenes. J. Vis. Comput. Animation **13**, 225–235 (2002)
6. Hoiem, D., Efros, A.A., Hebert, M.: Automatic photo pop-up. ACM Trans. Graph. (TOG) **24**, 577–584 (2005). ACM
7. Delage, E., Lee, H., Ng, A.Y.: A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2418–2428. IEEE (2006)
8. Saxena, A., Chung, S.H., Ng, A.Y.: Learning depth from single monocular images. In: Advances in Neural Information Processing Systems, pp. 1161–1168 (2005)
9. Saxena, A., Chung, S.H., Ng, A.Y.: 3-d depth reconstruction from a single still image. Int. J. Comput. Vis. **76**, 53–69 (2008)
10. Liu, B., Gould, S., Koller, D.: Single image depth estimation from predicted semantic labels. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1253–1260. IEEE (2010)
11. Li, C., Kowdle, A., Saxena, A., Chen, T.: Towards holistic scene understanding: feedback enabled cascaded classification models. In: Advances in Neural Information Processing Systems, pp. 1351–1359 (2010)
12. Karsch, K., Liu, C., Kang, S.B.: Depthtransfer: depth extraction from video using non-parametric sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence (2014)
13. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 2341–2353 (2011)
14. Wang, Y., Wang, R., Dai, Q.: A parametric model for describing the correlation between single color images and depth maps. Signal Processing Letters **21** (2014)
15. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing via label transfer. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 2368–2382 (2011)
16. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vis. **42**, 145–175 (2001)
17. Saxena, A., Sun, M., Ng, A.Y.: Make3d: learning 3d scene structure from a single still image. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 824–840 (2009)