Towards Unified Depth and Semantic Prediction from a Single Image Supplementary Material

Peng Wang[1] Xiaohui Shen[2] Zhe Lin[2] Scott Cohen[2] Brian Price[2] Alan Yuille[1]
[1]University of California, Los Angeles [2]Adobe Research

Content of this supplementary material

1. *The generation of larger segments in Sec. 4.2.*
2. *Geometric-preserving cropping for data augmentation in Sec. 6.*
3. *The details about the weight matrix in Sec.2 and Eqn. (5).*
4. *Our semantic label mapping in the experiment section.*
5. *Additional qualitative results.*
6. *A video giving an application demo of lens blur.*

1. Generating larger segments for context.

In Sec. 4.2, to reduce ambiguity of local region appearance, we perform semantic and depth regression over larger segments s_L, instead of directly on superpixels. For generating the segments, we consider the multi-level segmentation, which is an important strategy to obtain different level of context. To generate compact, semantic meaningful segments, we consider the information including the metric from appearance, semantic edges and spatial relationships. Formally, kernel k-means is performed to cluster the superpixels $\{s_i\}_{i=1}^N$ into multiple levels, consisting of $30, 50, 100$ segments in our experiments (as stated in Sec. 4.2), using the same distance metric as in Eqn. (4),

$$\text{dist}_a(s,t) = \text{dist}_g(s,t) + \lambda_a \|\mathbf{f}_s - \mathbf{f}_t\|, \text{ where } dist_g(s,t) = \min_{\mathcal{P}\in\mathcal{P}_{st}} \sum_{p\in\mathcal{P}} d_e(p, p+1). \tag{1}$$

$dist_g(s,t)$ is the minimum distance over all possible path $\mathcal{P}_{st}$ from superpixel s to t. Here, $d_e(p, p+1)$ is the summed edge map value (produced by by [2]) over the connecting boundary of the two adjacent superpixels p and $p+1$. $\mathbf{f}_s$ is a 12 dimensional vector describing the local appearance feature of the superpixel s. It is composed of the mean and covariance of the pixels rgb values inside the superpixel s.

2. Geometric-preserving cropping for RGB-D images

Cropping the image is a major data augmentation strategy to generate more training data for the CNN learning. However in depth estimation, as stated in [1], cropping does not preserve the geometry property of the original depth image. In [1], the method divided the original depth by the value s which is the percentage of the cropped image area over the entire image. However, this operation also does not respect the geometry and might bring further distortion.

Here, we propose a geometric-preserving cropping strategy in which the new depth values in the cropped regions keep the original geometry properties.

Proposition 1. *With known camera parameter, given a RGB-D image, for the image cropping operation keeping the aspect ratio, the new camera center and the depth after cropping can be inferred.*

Proof: As showed in Fig. 1, cropping operation is equivalent to moving the camera closer with no rotation. Here, for simplicity, we use a normalized camera matrix for this proof, i.e. $\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f} \end{pmatrix}$, and assume we know the focal length f. It is easy to extend this approach to other general camera matrix cases. For a pixel $\mathbf{x} = \{u, v\}$ in an image, given its depth d, we are able to get the 3D coordination of the pixel in the camera coordinate system, i.e. $\mathbf{x}^3 = d\frac{\mathbf{v_x}}{\|\mathbf{v_x}\|} = \{x_1, x_2, x_3\}$, where $\mathbf{v_x} = \{u, v, f\}$ is in the ray direction from camera center to the pixel. After cropping, suppose the new camera center is translated to $\mathbf{c} = \{c_1, c_2, c_3, 1\}^T$ in the original camera coordinate system, which is the unknown variables. Meanwhile, on the cropped image, we know the new location of the pixel $\mathbf{x}$, which is $\mathbf{x}' = \{u\prime, v\prime\}$. Then, the new camera center $\mathbf{c}$ should locate at the line passing point $\mathbf{x}^3$ with the direction $\{u\prime, v\prime, f\}$. This constraint provides two equations for the new camera center $\mathbf{c}$ as the collinearity of $\mathbf{c}$ and $\mathbf{x}^3$ withholds, i.e.

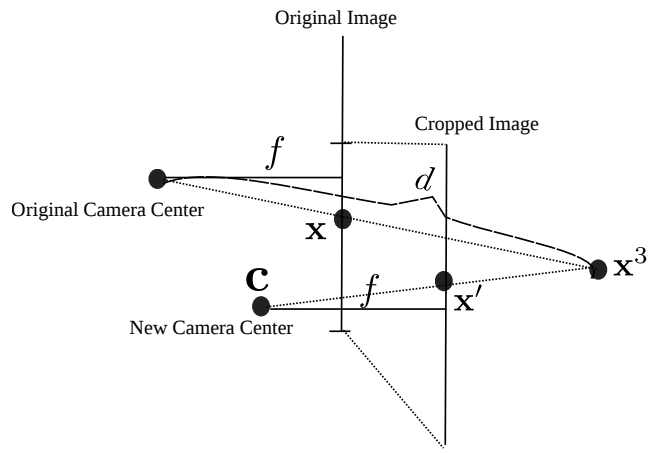

Figure 1. Ilustration of the proof.

$$\frac{c_1 - x_1}{u\prime} = \frac{c_2 - x_2}{v\prime} = \frac{c_3 - x_3}{f} \quad \Rightarrow \mathbf{A_x c} = 0; \text{ where } \mathbf{A_x} = \begin{pmatrix} \frac{1}{u\prime} & -\frac{1}{v\prime} & 0 & -\frac{x_1}{u\prime} + \frac{x_2}{v\prime} \\ \frac{1}{u\prime} & 0 & -\frac{1}{f} & -\frac{x_1}{u\prime} + \frac{x_3}{f} \end{pmatrix}$$

Thus, by considering all the pixels within the cropped image, we can find the optimal $\mathbf{c}$ through minimizing $\mathbf{Ac} = \mathbf{0}$ where $\mathbf{A}$ is the matrix concatenating all the constraints from pixels. This is a standard least square problem and can be solved through SVD or any matrix decomposition method. In practical, we can solve this simply by evenly sampling 200 pixels from the cropped image.

At last, in data augmentation for the global CNN with joint depth and semantic training, in addition to the crop size of $[228, 304]$, we add one extra crop size $[261, 196]$, and resize the cropped images to $[228, 304]$.

3. Weight matrix for the edge potential

In Eqn. (5), we have a semantic weight $w(l_s, l_t)$ as a function weighting the depth smoothness between two adjacent segments s, t when their semantic labels are l_s, l_t. All the semantic weights form a weight matrix $\mathbf{W}$, which is a $k \times k$ matrix where k is the number of semantic labels ($k = 5$ in our case). The intuition is that the depth smoothness between two adjacent segments should vary in terms of their semantic labels. For example, depth smoothness might be highly required for the segment pairs within ground regions, but might not be necessary for adjacent segments between ground and object regions.

In our approach, we propose to learn the matrix $\mathbf{W}$ through sampling adjacent segment pairs with different semantic labels. For getting the weight of a particular semantic label pair, we compute the average depth gradient value along the overlapping edge from all the corresponding pairs. Formally, the weight $w(l_m, l_n)$ is computed as,

$$w(l_m, l_n) = exp\left\{ -\left(\frac{1}{N} \sum_{j=1}^{N} \sum_{i \in s_j \cap t_j} |\nabla \mathbf{D}_j(i)| \right) / \sigma_w \right\}$$

where s_j, t_j are the j_{th} superpixel pair that having $l_{s_j} = l_m$ and $l_{t_j} = l_n$ or the other way around. N is number of the superpixel pairs. i is the pixel index. $\mathbf{D}_j$ is the depth map including the superpixel pair s_j, t_j. σ_w is a scale parameter which is set to be the average overlapping boundary length of all the superpixels pairs. We set the weight within the same class as one except the class of objects as its high variance. Some label pairs never appear in two adjacent segments in the training data (e.g., Ground and Ceiling). For the weight of these pairs, we set it to be a very large value $w_{max} = 1000$ to avoid labelling two adjacent segments with those label pairs that rarely occur.

Finally, after computed the score, we normalize the weight in the range of $[0.2, 1]$ without considering w_{max}, resulting in a final weight matrix as,

	Ground	Vertical	Ceiling	Furnitures	Objects
Ground	1	0.812	1000	0.453	0.335
Vertical	0.812	1	0.891	0.719	0.651
Ceiling	1000	0.891	1	1000	0.2
Furnitures	0.453	0.719	1000	1	0.412
Objects	0.335	0.651	0.2	0.412	0.4

In the weight matrix, we see a high smoothness is required for class among Ground, Vertical and Ceiling, while a relative low weight for class between Ground and Objects.

4. The semantic label mapping

The following table shows the label mapping from the original semantic labels in the NYU v2 dataset to our defined five labels.

Our Label	Original Label
Ground	floor, floor mat
Vertical	classroom board, blinds, window cover, door frame, door curtain, wardrobe, garage door, cabinet, wall, wall decoration, wall stand, dishwasher, door, projector screen, mirror, mailshelf, bookshelf, storage shelvesbooks, refrigerator
Ceiling	ceiling,roof
Furnitures	table, desk, bed, coffee table, mattress, sofa
Objects	All the rest

5. Additional qualitative results

In Fig. 2, we show more comparison results with the two state-of-the-art methods, i.e. DC Depth [3] and Depth CNN [1]. In Fig. 3 and Fig. 4, we show additional qualitative results by our algorithm including the semantic and depth output from the global CNN, and the joint HCRF combing both global and local inference. We can see that the results of joint HCRF contain much more fine-level details and structures than the global results.

6. An application demo of depth filter for lens blur

In the attached video, we provide a demo showing a potential application of our estimated depth map, i.e. depth filter for lens blur. At Left-top, we show the original image, and at bottom row, we show our segmented results and estimated depth map. By clicking at the focusing location on the original image, we show the lens blurred results at the right top. Users may generate their favourite results by choosing different focusing locations or selecting different sizes of the blur kernels.

We also uploaded a high quality video onto YouTube at https://www.youtube.com/watch?v=v6---biFPds&feature=youtu.be in case you can not open the video.

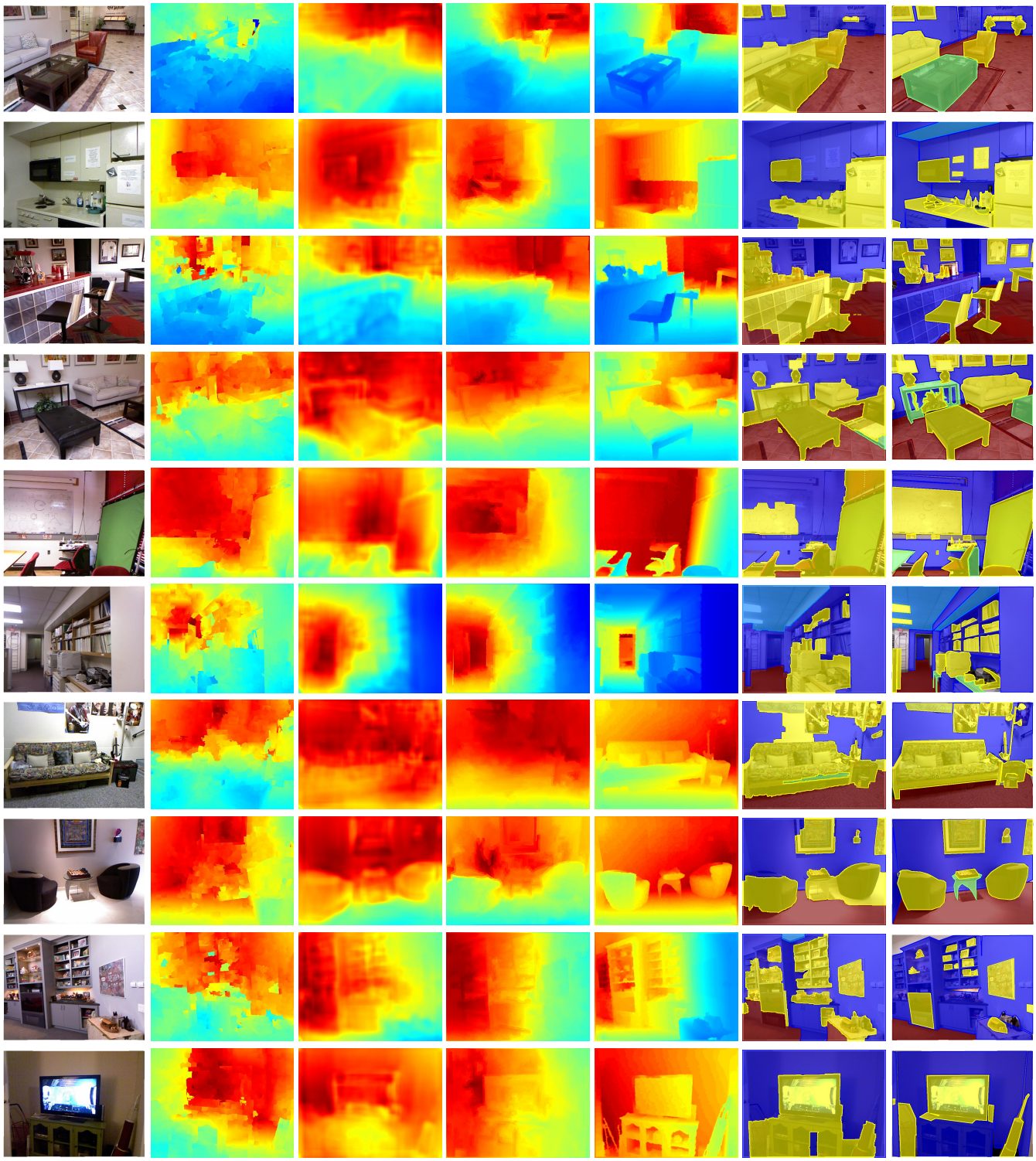

Figure 2. Additional qualitative comparison with other approaches. Depth maps are normalized by respective max depth (Best view in color).

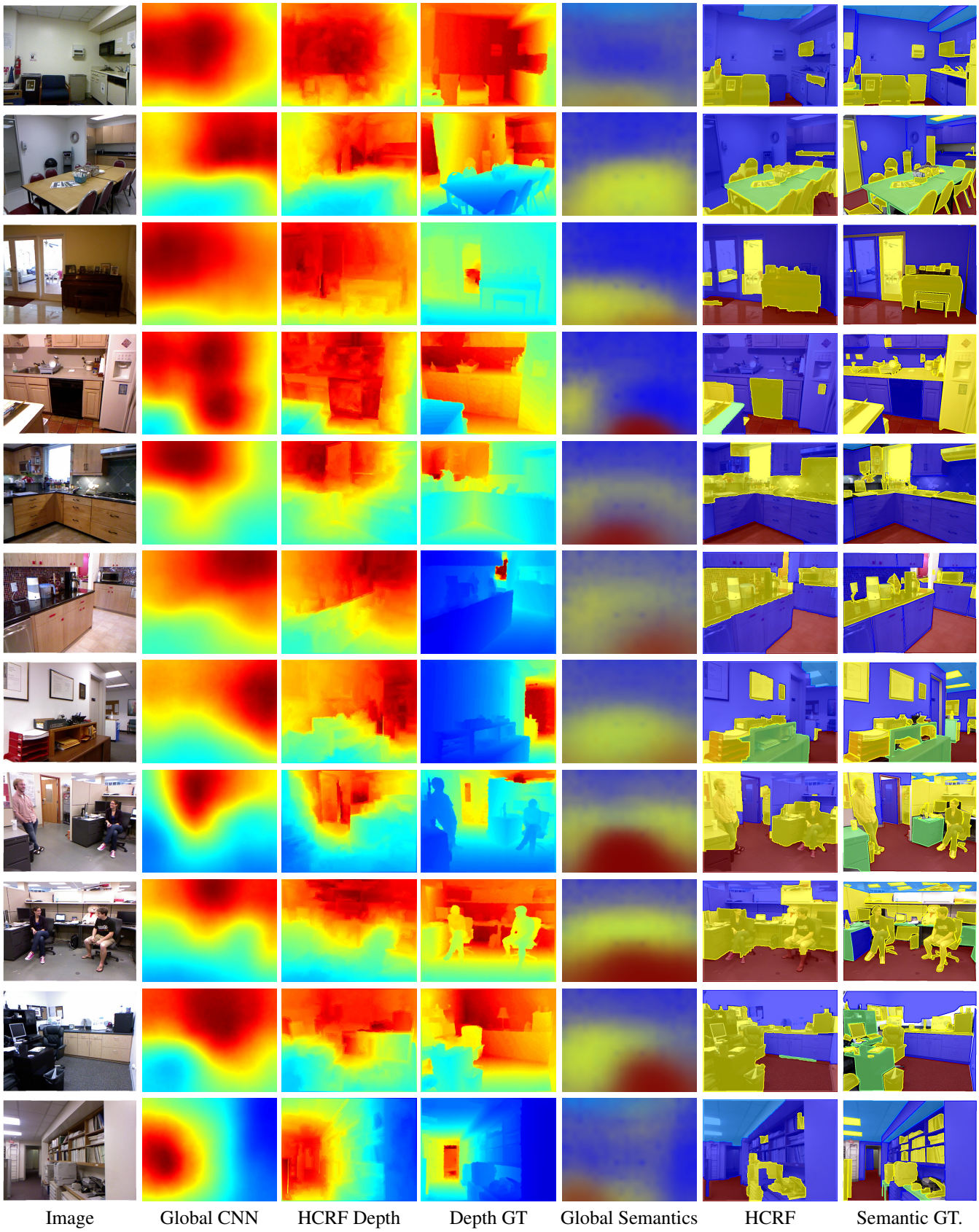

Figure 3. Additional qualitative results.

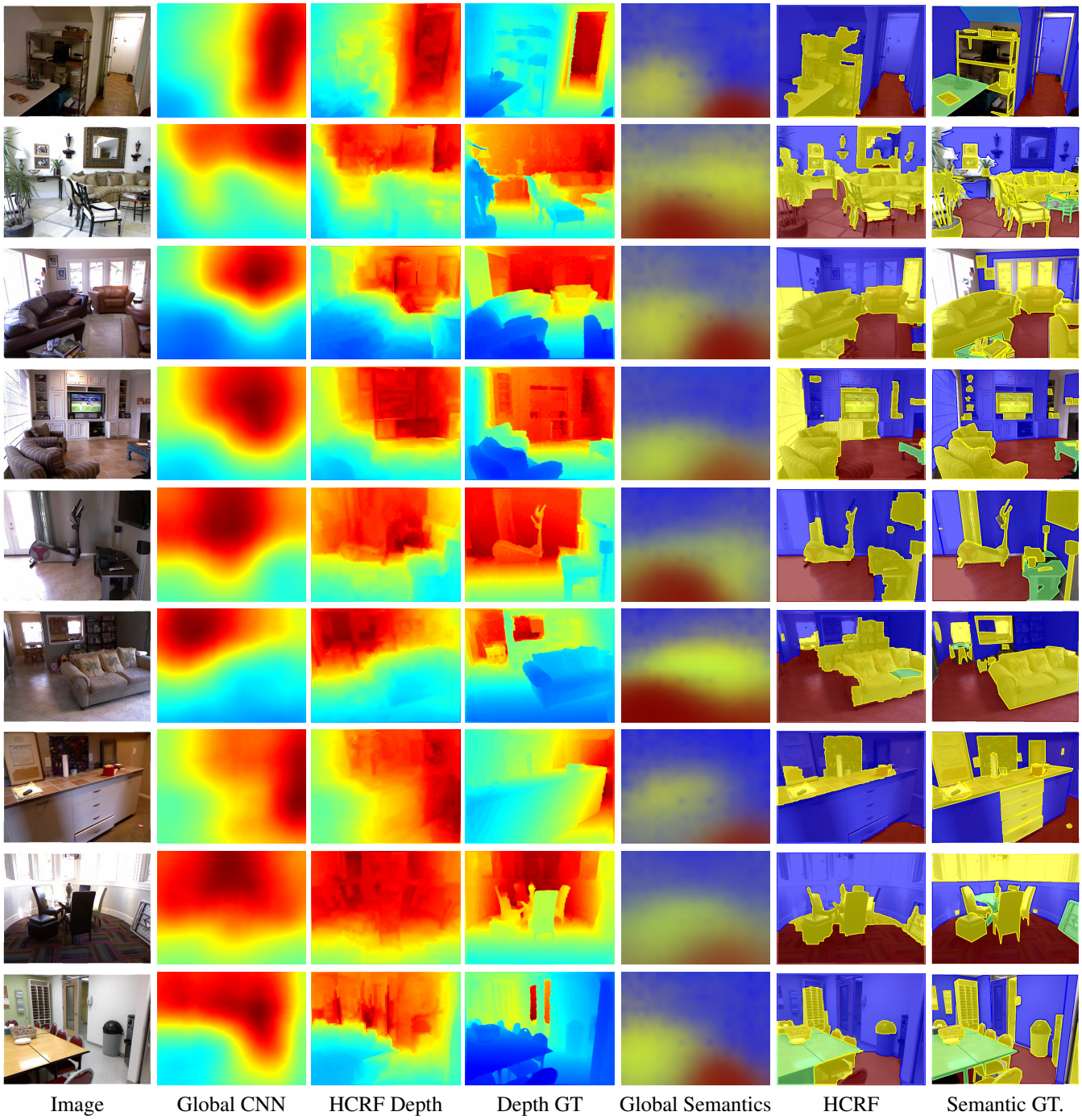

Figure 4. Additional qualitative results.

References

[1] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*. 2014. 1, 3, 4

[2] M. Leordeanu, R. Sukthankar, and C. Sminchisescu. Efficient closed-form solution to generalized boundary detection. In *ECCV (4)*, pages 516–529, 2012. 1

[3] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *CVPR*, June 2014. 3, 4