Pattern-Affinitive Propagation across Depth, Surface Normal and Semantic Segmentation

Zhenyu Zhang[1†] Zhen Cui[1†*] Chunyan Xu[1†]
Yan Yan[1†] Nicu Sebe[2‡] Jian Yang[1†*]

[1]PCA Lab, Nanjing University of Science and Technology
zhangjesse, zhen.cui, cyx, yyan, csjyang@njust.edu.cn

[2]Multimedia and Human Understanding Group, University of Trento
niculae.sebe@unitn.it

Abstract

In this paper, we propose a novel Pattern-Affinitive Propagation (PAP) framework to jointly predict depth, surface normal and semantic segmentation. The motivation behind it comes from the statistic observation that pattern-affinitive pairs recur much frequently across different tasks as well as within a task. Thus, we can conduct two types of propagations, cross-task propagation and task-specific propagation, to adaptively diffuse those similar patterns. The former integrates cross-task affinity patterns to adapt to each task therein through the calculation on non-local relationships. Next the latter performs an iterative diffusion in the feature space so that the cross-task affinity patterns can be widely-spread within the task. Accordingly, the learning of each task can be regularized and boosted by the complementary task-level affinities. Extensive experiments demonstrate the effectiveness and the superiority of our method on the joint three tasks. Meanwhile, we achieve the state-of-the-art or competitive results on the three related datasets, NYUD-v2, SUN-RGBD and KITTI.

1. Introduction

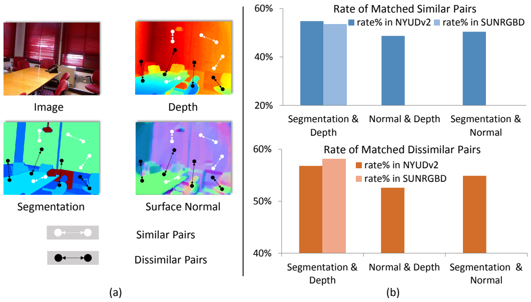


Figure 1. Statistics of matched affinity (or dissimilar) pairs across depth, surface normal and segmentation maps. (a) Visual exhibition. The point pairs colored white are the matched affinity pixels across three tasks at the same positions, while the pairs of black points correspond to dissimilar pixels across three maps. For the similarity metrics, REL/RMSE/Label consistency are taken respectively for the three maps. (b) Statistical results. We compute the success ratio of pairs matching across different maps on NYUD-v2 and SUN-RGBD datasets, and observe that the success ratios of pairs matching cross tasks are rather high.

The predictions of depth, surface normal and semantic segmentation are important and challenging for scene understanding. Also, they have many potential industrial applications such as autonomous driving system [4], simultaneous localization and mapping (SLAM) [52] and socially interactive robotics [12]. Currently, most methods [10, 11, 13, 14, 40, 43] focused on one of the three tasks, and they also achieved the state-of-the-art performance through the technique of deep learning.

In contrast to the single-task methods, recently, several joint-task learning methods [58, 62, 46, 32] on these tasks have shown a promising direction to improve the predictions by utilizing task-correlative information to boost for each other. In a broad sense, the problem of joint-task learning has been widely studied in the past few decades [3]. But more recently most approaches took the technique line of deep learning for possible different tasks [41, 16, 18, 25, 26]. However, most methods aimed to perform feature fu-

*Corresponding authors

†Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan and Jian Yang are with PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology. Zhenyu Zhang is also a visiting student in University of Trento.

‡Nicu Sebe is the head of Dept. of Information Engineering and Computer Science Leader of Multimedia and Human Understanding Group (MHUG) University of Trento.

sion or parameter sharing for task interaction. The fusion or sharing ways may utilize the correlative information between tasks, but there exist some drawbacks. For examples, the integration of different features might result into the ambiguity of information; the fusion does not explicitly model the task-level interaction where we do not know what information are transmitted. Conversely, could we find some explicitly common patterns across different tasks for the joint-task learning?

We take the three relative tasks: depth estimation, surface normal prediction and semantic segmentation, and then conduct a statistical analysis on those second-order patterns across different tasks on NYUD-v2 [49] and SUN-RGBD [51] dataset. First, we define the metric of any two pixels in the predicted images. The average relative error (REL) is used for depth images, the root mean square error (RMSE) is used for surface normal images, and the label consistency is for segmentation images. A pair of pixels have an affinity (or similar) relationship when their error is less than a specified threshold, otherwise they have a dissimilar relationship. Next, we accumulate the matching number of those similar pairs (or dissimilar pairs) with the same space positions across the three types of corresponding images. As shown in Fig. 1(a), the affinity pairs (colored white points) at the common positions may exist in different tasks. Meantime, there exist some common dissimilar pairs (colored black points) across tasks. The statistical results are shown in Fig. 1(b), where REL threshold of depth is set to 20%, and RMSE threshold of surface normal is set to 26% according to the performances of some state-of-the-art works [46, 1, 29]. We can observe that the success ratios of matching pairs across two tasks are rather high, and around 50% - 60% similar pairs are matched. Moreover, we have the same observation on the matching dissimilar pairs, where REL threshold of depth is set to 20%, and RMSR threshold of surface normal is set to 40%. Anyhow, this observation of the second-order affinities is great important to bridge two tasks.

Just motivated by the statistical observation, in this paper we propose a Pattern-Affinitive Propagation (PAP) framework to utilize the cross-task affinity patterns to jointly estimate depth, surface normal and semantic segmentation. In order to encode long-distance correlations, the PAP utilizes non-local similarities within each task, different from the literatures [39, 5] only considering local neighbor relationships. These pair-wise similarities are formulated as an affinity matrix to encode the pattern relationships of the task. To spread the affinity relationships, we take two propagation stages, cross-task propagation and task-specific propagation. The affinity relationships across tasks are first aggregated and optimized to adapt to each specific task by calculating on three affinity matrices. We then conduct an iterative task-specific diffusion on each task by leveraging the optimized affinity information from the corresponding other two tasks. The diffusion process is performed in the feature space so that the affinity information of other tasks can be widely spread into the current task. Finally, the learning of affinitive patterns and the two-stage propagations are encapsuled into an end-to-end network to boost the prediction process of each task.

In summary, our contributions are in three aspects: i) Motivated by an observation that pattern-affinitive pairs recur much frequently across different tasks, we propose a novel Pattern-affinitive Propagation (PAP) method to utilize the matched non-local affinity information across tasks. ii) Two-stage affinity propagations are designed to perform cross-task and task-specific learning. An adaptive ensemble network module is designed for the former while the strategy of graph diffusion is used for the latter. iii) We make extensive experiments to validate the effectiveness of PAP method and its modules therein, and achieve the competitive or superior performances on depth estimation, surface normal prediction and semantic segmentation on NYUD-v2 [49], SUN-RGBD [51], and KITTI [53] datasets.

2. Related Works

Depth Estimation: Many works have been proposed for monocular depth estimation [10, 11, 37, 32, 42, 29, 63, 54, 47, 60, 58, 46, 62]. Recently, Xu *et al.* [59] employed multi-scale continuous CRFs as a deep sequential network for depth prediction. Fu *et al.* [15] tried to consider the ordinal information in depth maps and designed a ordinal regression loss function.

RGBD Semantic Segmentation: As the large RGBD dataset was released, some approaches [17, 21, 48, 8, 22, 34] attempted to fuse depth information for better segmentation. Recently, Qi *et al.* [45] designed a 3D graph neural network to fuse the depth information for segmentation. Cheng *et al.* [6] computed the important locations from RGB images and depth maps for upsampling and pooling.

Surface Normal Estimation: Recent methods designed for surface normal estimation are mainly based on deep neural networks [13, 14, 61, 55]. Wang *et al.* [56] designed a network to incorporate local, global and vanishing point information for surface normal prediction. In work of [1], a skip-connected architecture was proposed to fuse features from different layers for surface normal estimation. 3D geometric information was also utilized in [46] to predict depth and normal maps.

Affinity Learning: Many affinity learning methods were designed based on physical nature of the problems [19, 28, 30]. Liu *et al.* [38] improve the modeling of pair-wise relationships by incorporating many priors into diffusion process. Recently, work of [2] proposed an convolutional random walk approach to learn the image affinity by supervision. Wang *et al.* [57] proposed a non-local neu-

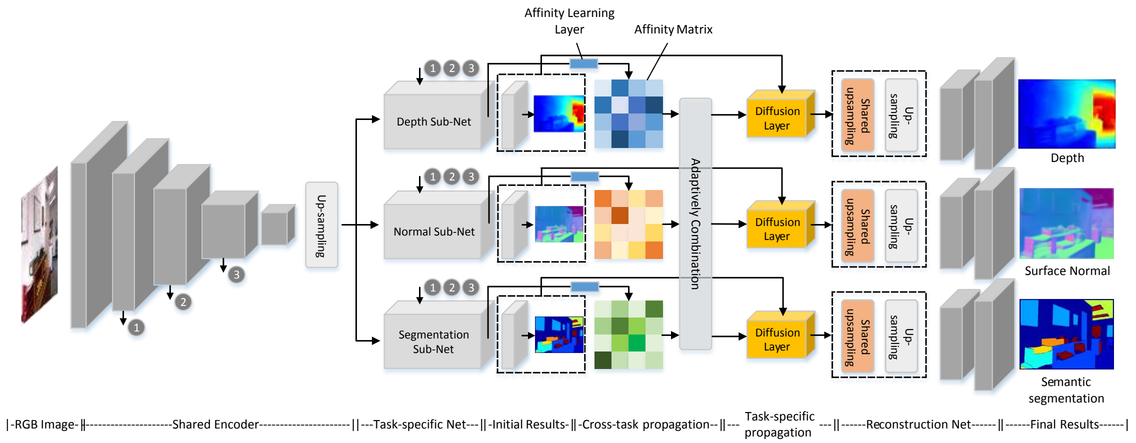

Affinity Learning Layer
Affinity Matrix
1 2 3
Depth Sub-Net
Normal Sub-Net
Segmentation Sub-Net
Up-sampling
1
2
3
Adaptively Combination
Diffusion Layer
Shared upsampling
Up-sampling
Depth
Surface Normal
Semantic segmentation
|-RGB Image-||--------------------Shared Encoder--------------------||---Task-specific Net---||-Initial Results-||-Cross-task propagation--||--- Task-specific propagation ---||------Reconstruction Net------||------Final Results------|

Figure 2. The overview of our Pattern-Affinitive Propagation network for jointly predicting depth, surface normal and semantic segmentation. The initial predictions are produced from each task-specific network. During cross-task propagation, the network firstly learns an affinity matrix by affinity learning layer to represent the pair-wise relationships of each task, then adaptively combines these matrices to propagate the cross-task affinitive patterns. Note that, the combined affinity matrices is different for each task. Then we use the combined matrix to conduct task-specific propagation by a diffusion layer, propagating the affinitive patterns back to the features for each task. Finally the diffused features are applied to three reconstruction networks to produce the final results with higher resolution.

ral network to mine the relationships with long distances. Some other works [39, 5, 23] tried to learn local pixel-wise affinity for semantic segmentation or depth completion. Our method is different from these approaches in the following aspects: needs no prior knowledge and is data-driven; needs no task-specific supervisons; learns the non-local affinity rather than limited local pair-wise relationships; learns the cross-task affinity information rather than learning the single-task affinity for task-level interaction.

3. Non-Local Affinities

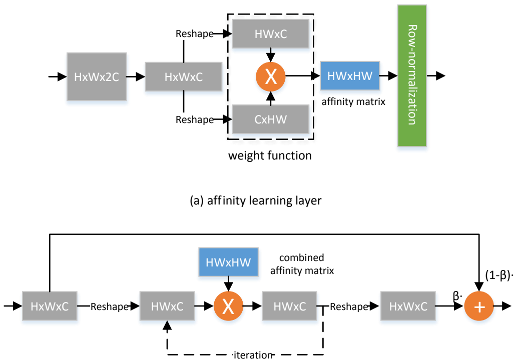
Our aim is to model the affinitive patterns among tasks, and utilize such complementary information to boost and regularize the prediction process of each task. According to our analysis aforementioned, we want to learn the pair-wise similarities and then propagate the affinity information into each task. Instead of learning local affinities as literature [39, 5], we attempt to utilize non-local affinities, which also recur frequently as illustrated in Fig. 1. Formally, suppose $\mathbf{x}_i, \mathbf{x}_j$ are the feature vectors of the i-th and j-th positions, we can define their similarity $s(\mathbf{x}_i, \mathbf{x}_j)$ through some functions such as L1 distance $\|\mathrm{x}_i - \mathrm{x}_j\|$, inner product $\mathbf{x}_i^T \mathbf{x}_j$, and so on. We employ the exponential function ($e^{s(\cdot,\cdot)}$ or $e^{-s(\cdot,\cdot)}$) to make the similarities non-negative and larger for those similar pairs than dissimilar pairs. To reduce the influence of scale, we normalize the similarity matrix $\mathbf{M}$ into $\mathbf{M}_{ij}/\sum_k \mathbf{M}_{ik}$, where $\mathbf{M}$ is the matrix of pair-wise similarities across all pixel positions. In these ways, the matrix $\mathbf{M}$ is **symmetric**, has **non-negative** elements and **finite Frobenius norm**. Accordingly, for the three tasks, we can compute their similarity matrices $\mathbf{M}_{\text{depth}}, \mathbf{M}_{\text{seg}}, \mathbf{M}_{\text{normal}}$ respectively. According to the above statistic analysis, we can propagate the affinities by integrating the three similarity matrices for one specific task, which will be introduced in the following section.

4. Pattern-Affinitive Propagation

In this section, we introduce the proposed Pattern-Affinitive Propagation (PAP) method. We efficiently implement the PAP method into a deep neural network through designing a series of network modules. The details are introduced in the following.

4.1. The Network Architecture

We implement the proposed method into a deep network as shown in Fig. 2, which depicts the network architecture. The RGB image is firstly fed into a shared encoder (e.g., ResNet [20]) to generate hierarchical features. Then we upsample the features of the last convolutional layer and feed them to three task-specific networks. Note that we also integrate multi-scale features derived from different layers of encoder with each task-specific network, as shown by the gray dots. Each task-specific network has two residual blocks, and produces the initial prediction after a convolutional layer. Then we conduct cross-task propagations to learn the task-level affinitive patterns. Each task-specific network firstly learns an affinity matrix by the affinity learning layer to capture the pair-wise similarities for each task, and secondly adaptively combine the matrix with other two affinity matrices to integrate the task-correlative information. Note that, the adaptively combined matrix is different for each task. After that, we conduct task-specific propagation via a diffusion layer to spread the learned affinitive patterns back to the feature space. In each diffusion process, we diffuse both initial prediction and the last features from each task-specific network by the combined affinity matrix.



(a) affinity learning layer

(b) diffusion process

Figure 3. The detailed information of affinity learning layer and diffusion process, and each block describes the feature and its shape. $\otimes$ represents the matrix multiplication. (a) affinity learning layer. The dashed box is corresponding to the function for computing similarities, and we only illustrate the dot-product as an example. (b) diffusion process. $\oplus$ represents the weighted sum with a parameter β. The dashed arrows are only performed when the iteration is not finished.

Finally, the diffused features of each task are fed into a reconstruction network to produce final prediction with higher resolution. We firstly use a shared and a task-specific upsampling block to upscale the feature maps. Each upsampling block is built as a up-projection block [29], and parameters in the shared upsampling block are shared for every task to capture correlative local details. After the upsampling with the two blocks, the features are concatenated and fed into a residual block to produce final predictions. The scale factor of each upsampling block is set to 2, and the final predictions are half of the input scale. This means that the number of upsampling blocks depends on the scale on which we want to learn affinity matrix. In experiments, we learn affinity matrices on 1/16, 1/8 and 1/4 input scale, which means there are 3, 2 and 1 upsampling stages in the reconstruction network respectively. The whole network can be trained in an end-to-end manner, and the details of the cross-task and task-specific propagations will be introduced in the following sections.

4.2. Cross-Task Propagation

In this section we elaborate how to conduct cross-task propagation. Firstly, we learn an affinity matrix by affinity learning layer to represent the pair-wise similarities for each task. The detailed architecture of the affinity learning layer can be observed in Fig. 3(a). Assuming the feature generated by the last layer of each task-specific network is $\mathrm{F} \in \mathbb{R}^{H\times W\times 2C}$, we firstly shrink it using a 1×1 convolutional layer to get the feature $\tilde{\mathrm{F}} \in \mathbb{R}^{H\times W\times C}$. Then $\tilde{\mathrm{F}}$ is reshaped to $\mathbf{X} \in \mathbb{R}^{HW\times C}$. We utilize matrix multiplication to compute pair-wise similarities of inner product, and obtain the affinity matrix $\mathbf{M} = \mathbf{X}\mathbf{X}^{\intercal} \in \mathbb{R}^{HW\times HW}$. Other pair-wise functions such as $e^{-\|\mathrm{X}_i - \mathrm{X}_j\|}$ can also be used, just not shown in the figure. Note that, different from non-local blocks [57], our affinity matrix must satisfy the symmetric and nonnegative properties to represent the pair-wise similarities. Finally, as each row of the matrix $\mathbf{M}$ represents the pair-wise relationships between one position and all other positions, we conduct normalization along each row of $\mathbf{M}$ to reduce the influence of scale. In this way, the task-level patterns can be represented in each $\mathbf{M}$. Note that we add no supervision to learn $\mathbf{M}$ as literature [2], because such supervision will cost extra memories and be not easy to define for some tasks. After that, we want to integrate the cross-task information for each task. Denote these three tasks as T_1, T_2, T_3, and the corresponding affinity matrices as $\mathbf{M}_{T_1}\mathbf{M}_{T_2}\mathbf{M}_{T_3}$, then we can learn weights $\alpha_k^{T_i}$ $(k = 1, 2, 3, \sum_{k=1}^{n} \alpha_k^{T_i} = 1)$ to adaptively combine the matrices as:

$$\hat{\mathbf{M}}_{T_i} = \alpha_1^{T_i} \cdot \mathbf{M}_{T_1} + \alpha_2^{T_i} \cdot \mathbf{M}_{T_2} + \alpha_3^{T_i} \cdot \mathbf{M}_{T_3}. \quad (1)$$

In this way, the cross-task affinitive patterns can be propagated into $\hat{\mathbf{M}}_{T_i}$. In practice, we implement affinity learning layers at decoding process on 1/16, 1/8 and 1/4 input scale respectively, hence it actually learns non-local patch-level relationships.

4.3. Task-Specific Propagation

After obtaining the combined affinity matrices, we spread such affinitive patterns into the feature space of each task by the task-spacific propagation. Different from non-local block [57] and local spatial propagation [39, 5], we perform an iterative non-local diffusion process in each diffusion layer to capture long-distance similarities, as illustrated in Fig. 3(b). The diffusion process is performed on initial prediction as well as features from task-specific network. Without loss of generality, assuming feature or initial prediction $\mathrm{P} \in \mathbb{R}^{H\times W\times C}$ is from task-specific network, we firstly reshape it to $\mathbf{h} \in \mathbb{R}^{HW\times C}$, and perform one step diffusion by using matrix multiplication with $\hat{\mathbf{M}}$. In this way, the feature vector of each position is obtained by weighted accumulating feature vectors of all positions using the learned affinity. Note that such one-step diffusion may not deeply and effectively propagate the affinity information to the feature space, we perform the multi-step iterative diffusion as:

$$\mathbf{h}^{t+1} = \hat{\mathbf{M}}\mathbf{h}^t, \;\; t \geq 0, \quad (2)$$

where $\mathbf{h}^t$ means the diffused feature (or prediction) at step t. Such diffusion process can be also expressed with a partial differential equation (PDE):

$$\begin{aligned} \mathbf{h}^{t+1} &= \hat{\mathbf{M}}\mathbf{h}^t = (\mathbf{I} - \mathbf{L})\mathbf{h}^t, \\ \mathbf{h}^{t+1} - \mathbf{h}^t &= -\mathbf{L}\mathbf{h}^t, \\ \partial_t \mathbf{h}^{t+1} &= -\mathbf{L}\mathbf{h}^t, \end{aligned} \quad (3)$$

where $\mathbf{L}$ is the Laplacian matrix. As $\hat{\mathbf{M}}$ is normalized and has finite Frobenius norm, the stability of such PDE can be guaranteed [39]. Assuming we totally perform t^* steps in each diffusion layer, in order to prevent the feature deviating too much from the initial one, we use the weighted accumulation on the initial feature (or prediction) $\mathbf{h}^0$ as:

$$\mathbf{h}^{\text{out}} = \beta \mathbf{h}^{t^*} + (1-\beta)\mathbf{h}^0, \ \ 0 \leq \beta \leq 1, \qquad (4)$$

where $\mathbf{h}^{\text{out}}$ means the final output from a diffusion layer. In this way, the learned affinitive patterns in each $\hat{\mathbf{M}}_{T_i}$ can be effectively propagated into each task T_i.

4.4. The Loss Function

In this section we introduce a pair-wise affinity loss for our PAP network. As PAP method is designed to learn task-correlative pair-wise similarities, we also hope our loss function can enhance the pair-wise constraints. Firstly we define the prediction at position i is $\hat{z}_i$, and the corresponding ground truth is z_i. Then we define the pair-wise distance in prediction and corresponding ground truth as $\hat{d}_{ij} = |\hat{z}_i - \hat{z}_j|$ and $d_{ij} = |z_i - z_j|$. We hope the distance in prediction to be similar to ground truth, so the pair-wise loss can be defined as $\mathcal{L}_{\text{pair-wise}} = \sum_{\forall i,j} |\hat{d}_{ij} - d_{ij}|$. As the calculation of the pair-wise loss in each task will have a high memory burden, so we randomly select S pairs from each task and then compute the pair-wise loss $\mathcal{L}_{\text{pair-wise}} = \sum_{S} |\hat{d}_{ij} - d_{ij}|$. As the pairs are randomly selected, such pair-wise loss can capture similarities of various-distance pairs, not only the adjacent pixels in [10]. Meanwhile, we also use berHu loss [29], L1 loss and cross-entropy loss for depth estimation, surface normal prediction and semantic segmentation respectively, which are denoted as $\mathcal{L}^{T_i}$ (T_i means the i-th task). Finally the total loss of the joint task learning problem can be defined as:

$$\mathcal{L} = \sum_{T_i} \lambda_{T_i} (\mathcal{L}^{T_i} + \xi_{T_i} \mathcal{L}^{T_i}_{\text{pair-wise}}), \qquad (5)$$

where $\mathcal{L}^{T_i}_{\text{pair-wise}}$ is the pair-wise loss for the corresponding i-th task, and λ_{T_i} and ξ_{T_i} are two weights for the i-th task.

5. Experiment

5.1. Dataset

NYUD-v2: The NYUD v2 dataset [49] consists of RGB-D images of 464 indoor scenes. There are 1449 images with semantic labels, 795 of them are used for training and the remaining 654 images for testing. We randomly select more images (12k, same as [29, 62]) from the raw data of official training scenes. These images have the corresponding depth maps but no semantic labels or surface normals. We follow the procedure in [13] and [46] to generate surface normal ground truth. In this way, we can use more data to train our model for jointly depth and surface normal prediction.

SUN RGBD: The SUN RGBD dataset [51] contains 10355 RGBD images with semantic labels of which 5285 for training and 5050 for testing. We use the official training set with depth and semantic labels to train our network, and the official testing set for evaluation. There is no surface normal ground truth on this dataset, so we perform experiments on jointly predicting depth and segmentation on this dataset.

KITTI: KITTI online benchmark [53] is a widely-used outdoor dataset for depth estimation. There are 4k images for training, 1k images for validating and 500 images for testing on the online benchmark. As it has no semantic labels or surface normal ground truth, we mainly transform such information using our PAP method to demonstrate that PAP can distilling knowledge to improve the performance.

5.2. Implementation Details and Metrics

We implement the proposed model using Pytorch [44] on a single Nvidia P40 GPU. We build our network based on ResNet-18 and ResNet-50, and each model is pre-trained on the ImageNet classification task [7]. In diffusion process, we use a same subsampling strategy as [57] to downsample h in Eqn. (2), which can reduce the amount of pair-wise computation by 1/4. We set the trade-off parameter β to 0.05. 300 pairs are randomly selected to compute the pair-wise loss in each task. We simply set $\lambda_{T_i} = \frac{1}{3}$ and $\xi_{T_i} = 0.2$ to balance the loss functions. Initial learning rate is set to 10^{-4} for the pre-trained convolutional layers and 0.01 for the other layers. For NYUD-v2, we train the model of 795 training images for 200 epochs and fine-tune 100 epochs, and train the model of 12k training images for jointly depth/normal predicting for 30 epochs and fine-tune for 10 epochs. For SUN-RGBD dataset, we train the model for 30 epochs and fine-tune it for 30 epochs using a learning rate of 0.001. For KITTI, we first train the model on NYUD-v2 for surface normal estimation, and then freeze the surface normal branch to train depth branch on KITTI for 15 epochs, finally we freeze the normal branch and fine-tune the model on KITTI for 20 epochs.

Similar to the previous works [29, 10, 59], we evaluate our depth prediction results with the root mean square error (rmse), average relative error (rel), root mean square error in log space (rmse-log), and accuracy with threshold (δ): % of $\widetilde{x_i}$ s.t. max($\frac{\widetilde{x_i}}{x_i}$, $\frac{x_i}{\widetilde{x_i}}$)=$\delta$, $\delta = 1.25, 1.25^2, 1.25^3$, where $\widetilde{x_i}$ is the predicted depth value at the pixel i, n is the number of valid pixels and x_i is the ground truth. The evaluation metrics for surface normal prediction [56, 1, 10] are mean of angle error (mean), medians of the angle error (median), root mean square error for normal (rmse-n %), and pixel accuracy as percentage of pixels with angle error below threshold η where $\eta \in [11.25^\circ, 22.50^\circ, 30^\circ]$. For the evaluation of semantic segmentation results, we follow the recent works [6] [24] [35] and use the common metrics including pixel

Table 1. Analyses on Joint task learning on NYU Depth V2.

Metric	rmse	iou	rmse-n
Depth only	0.570	-	
Segmentation only	-	42.8	-
Normal only	-	-	28.7
Depth&Seg jointly	0.556	44.3	-
Depth&Normal jointly	0.550	-	28.1
Segmentation&Normal jointly	-	44.5	28.3
Three task jointly	**0.533**	**46.2**	**26.9**

Table 2. Comparisons of different network settings and baselines on NYU Depth v2 dataset.

Method	rmse	IoU	rmse-n
initial prediction	0.582	41.3	29.6
+ PAP w/o cross-t prop.	0.574	41.8	29.1
+ PAP cross-t prop.	0.558	43.1	28.5
+ PAP cross-t prop. + recon-net	0.550	43.8	28.2
+ PAP cross-t prop + recon-net + pair-loss	0.543	44.2	27.8
+ cross-stich [41]	0.550	43.5	28.2
+ CSPN [5]	0.548	43.8	28.0
aff-matrix on 1/16 input scale	0.543	44.2	27.8
aff-matrix on 1/8 input scale	0.533	**46.2**	26.9
aff-matrix on 1/4 input scale	**0.530**	46.5	**26.7**
Inner product	0.543	44.2	27.8
L1 distance	0.540	44.0	27.9

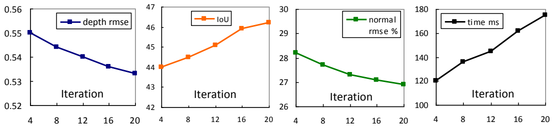


Figure 4. The influence of the iterations in diffusion process. The performance and time burden changes can be seen as a trade-off.

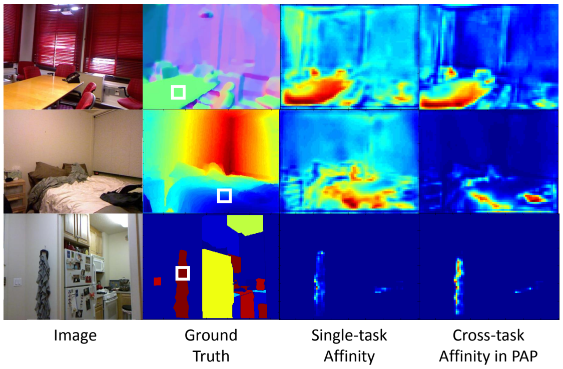


Figure 5. Visualization of the single-task and our cross-task affinity maps at the white point for each task. We can see that the pair-wise similarities at the white point can be improved and corrected in our PAP method.

accuracy (pixel-acc), mean accuracy (mean-acc) and mean intersection over union (IoU).

5.3. Ablation Study

In this section we perform many experiments to analyse the influence of different settings in our method.

Effectiveness of joint task learning: We first analyse the benefit of joint predicting depth, surface normal and semantic segmentation using our PAP method. The networks are trained on NYUD v2 dataset, and we select ResNet-18 as our shared network backbone and only learn the affinity matrix on 1/8 input scale in each experiment. As illustrated in Table 1, we can see that joint-task models gets superior performances than the single task model, and further jointly learning three tasks obtains best results. It can be revealed that our PAP method does boost each task in the jointly learning procedures.

Analysis on network settings: We perform many experiments to analyse the effectiveness of each network modules. In each experiment we use ResNet-18 as our network backbone for equally comparing, and each model is trained on NYUD v2 dataset for the three tasks. The result can be seen in Table 2. Note that the results of first five rows are computed from the model with affinity matrix learned on 1/16 input scale. We can observe that PAP, reconstruction net and pair-wise loss can all contribute to improve the performance. We also compare two approaches in the same settings, i.e., cross-stich units [41] and convolutional spatial propagation layers [5] which can also fuse and interact cross-task information. We find that they obtain weaker performances. It may be attributed to that: a) cross-stich layer only combines features, but cannot represent the affinitive patterns between tasks; b) they only use limited local information. The middle three rows of the Table 2 show the influence on which scale the affinity matrix is learned. We can find that learning affinity matrix on a larger scale may be beneficial, as the larger affinity matrices can describe the similarities between more patches. Note that the improvements of learning matrix on 1/4 input scale are comparatively smaller, and the reason may be that learning good non-local pair-wise similarities becomes more difficult with scale increasing. Finally we show the results using different functions to calculate the similarities. We find that these two functions does produce different performances, but with little difference. Hence, we mainly use dot product as our weight function in the following experiments for convenience.

Influence of the iteration: Here we make experiments to analyse the influence of the iterative steps in Eqn. (2). The models are based on ResNet-18 and trained on NYUD v2 dataset, and the affinity matrices are learned on 1/8 input scale. While testing, the input size is 480×640. As illustrated in Fig. 4, we can see that the performances of all tasks are improved with more iterations, at least in such a range. These results demonstrate that the pair-wise constraints and regularization may be enhanced with more iterations in diffusion. But the testing time will also increase with more steps, which can be seen as a trade-off.

Visualization of the affinity matrices: We show several

Table 3. Comparisons with the state-of-the-art depth estimation approaches on NYU Depth V2 Dataset.

Method	data	rmse	rel	log	δ_1	δ_2	δ_3
HCRF [32]	795	0.821	0.232	-	0.621	0.886	0.968
DCNF [37]	795	0.824	0.230	-	0.614	0.883	0.971
Wang [54]	795	0.745	0.220	0.262	0.605	0.890	0.970
NR forest [47]	795	0.744	0.187	-	-	-	-
Xu [60]	795	0.593	0.125	-	0.806	0.952	0.986
PAD-Net [58]	795	0.582	0.120	-	0.817	0.954	0.987
Eigen [11]	120k	0.877	0.214	0.285	0.611	0.887	0.971
MS-CNN [10]	120k	0.641	0.158	0.214	0.769	0.950	0.988
MS-CRF [59]	95k	0.586	0.121	-	0.811	0.954	0.987
FCRN [29]	12k	0.573	0.127	0.194	0.811	0.953	0.988
GeoNet [46]	16k	0.569	0.128	-	0.834	0.960	0.990
AdaD-S [42]	100k	0.506	**0.114**	-	**0.856**	0.966	0.991
DORN [15]	120k	0.509	0.115	-	0.828	0.965	0.992
TRL [62]	12k	0.501	0.144	0.181	0.815	0.962	0.992
Ours d+s+n	795	0.530	0.142	0.190	0.818	0.957	0.988
Ours d+n	12k	**0.497**	0.121	**0.175**	0.846	**0.968**	**0.994**

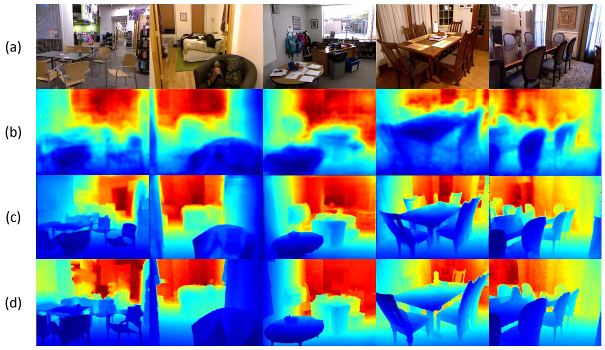


Figure 6. Visualization of our predicted depth maps. (a) image; (b) predictions of [60]; (c) our results; (d) ground truth. We can find that our predictions have obviously finer details and closer to ground truth.

Table 4. Comparisons with the state-of-the-art surface normal estimation approaches on NYU Depth V2 Dataset.

Method	mean	median	rmse-n	11.25°	22.50°	30°
3DP [13]	36.3	19.2	-	16.4	36.6	48.2
UNFOLD [14]	35.2	17.9	-	40.5	54.1	58.9
Discr. [61]	33.5	23.1	-	27.7	49.0	58.7
MS-CNN [10]	23.7	15.5	-	39.2	62.0	71.1
Deep3D [56]	26.9	14.8	-	42.0	61.2	68.2
SkipNet [1]	19.8	12.0	28.2	47.9	70.0	77.8
SURGE [55]	20.6	12.2	-	47.3	68.9	76.6
GeoNet [46]	19.0	11.8	26.9	48.4	71.5	79.5
Ours-VGG16	**18.6**	**11.7**	**25.5**	**48.8**	**72.2**	**79.8**

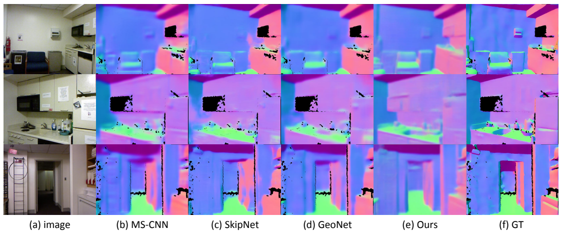


Figure 7. Visualization of our predicted surface normal. (a) image; (b) predictions of [10]; (c) predictions of [1] ; (d) predictions of [46]; (e) our results; (f) ground truth.

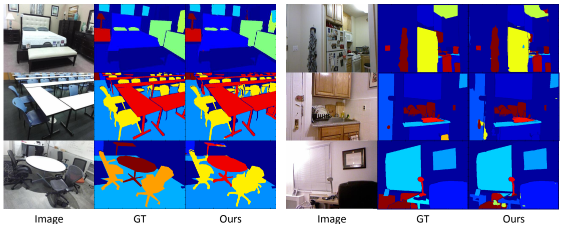


Figure 8. Qualitative semantic segmentation results of our method on NYUD-v2 and SUNRGBD datasets.

examples of the learned affinity maps in Fig. 5. Note that the affinity maps belong to the white point in each image. We can see that the single-task affinity maps often show improper pair-wise relationships, while the cross-task affinity maps in our PAP method have closer relationships with the points which have similar depth, normal direction and semantic label. As the affinity matrices is non-local and actually a dense graph, it can well represent the long-distance similarities. Such observations demonstrate that the cross-task complementary affinity information can be learned to refine the single-task similarities in PAP method. Though without supervision as [2], our PAP method can still learn good affinity matrices in such task-regularized unsupervised approach.

5.4. Comparisons with state-of-the-art methods

Depth Estimation: We mainly perform experiments on NYUD-v2 dataset to evaluate our depth predictions. The models are based on ResNet-50. As illustrated in Table 3, our model trained for three tasks (ours d+s+n) obtains competitive results, though only 795 images are used for training. Such results demonstrate that our PAP method can well boost each task and benefit joint task learning with limited training data. For the model trained for depth&normal prediction (ours d+n), with more training data can be used, our PAP method gets significantly best performances in most of the metrics with more training data, which well proves the effectiveness of our approach. Qualitative results can be observed in Fig. 6, compared with the recent work [60], our predictions are more fine-detailed and closer to ground truth.

Surface Normal Estimation: We mainly evaluate our surface normal predictions on NYUD-v2 dataset. As previous methods mainly build their network based on VGG-16 [50], we also utilize the same setting in our experiments. As illustrated in Table 4, our PAP method obtains obviously superior performances than the previous approaches in all metrics. Such results well demonstrate that our joint task learning method can boost and benefit the surface normal estimation. Qualitative results can be observed in Fig. 7, we can find that our method can produce better or competitive results.

RGBD Semantic Segmentation: We evaluate our segmentation results on widely-used NYUD-v2 and SUN-RGBD datasets. The model in each experiment is build

Table 5. Comparisons the state-of-the-art semantic segmentation methods on NYU Depth v2 dataset.

Method	data	pixel-acc	mean-acc	IoU
FCN [40]	RGB	60.0	49.2	29.2
Context [36]	RGB	70.0	53.6	40.6
Eigen *et al.* [10]	RGB	65.6	45.1	34.1
B-SegNet [24]	RGB	68.0	45.8	32.4
RefineNet-101 [35]	RGB	72.8	57.8	44.9
PAD-Net [58]	RGB	75.2	62.3	50.2
TRL-ResNet50 [62]	RGB	**76.2**	56.3	46.4
Deng *et al.* [8]	RGBD	63.8	-	31.5
He *et al.* [22]	RGBD	70.1	53.8	40.1
LSTM [34]	RGBD	-	49.4	-
Cheng *et al.* [6]	RGBD	71.9	60.7	45.9
3D-GNN [45]	RGBD	-	55.7	43.1
RDF-50 [48]	RGBD	74.8	60.4	47.7
Ours-ResNet50	RGB	**76.2**	**62.5**	**50.4**

Table 6. Comparison with the state-of-the-art semantic segmentation methods on SUN-RGBD dataset.

Method	data	pixel-acc	mean-acc	IoU
Context [36]	RGB	78.4	53.4	42.3
B-SegNet [24]	RGB	71.2	45.9	30.7
RefineNet-101 [35]	RGB	80.4	57.8	45.7
TRL-ResNet50 [62]	RGB	83.6	58.9	50.3
LSTM [34]	RGBD	-	48.1	-
Cheng *et al.* [6]	RGBD	-	58.0	-
CFN [9]	RGBD	-	-	48.1
3D-GNN [45]	RGBD	-	57.0	45.9
RDF-152 [48]	RGBD	81.5	**60.1**	47.7
Ours-ResNet50	RGB	**83.8**	58.4	**50.5**

Table 7. Comparison with the state-of-the-art methods on KITTI online benchmark (lower is better).

Method	SILog	sqErrRel	absErrRel	iRMSE	time
DORN [15]	11.77	2.23	8.78	12.98	0.5s
VGG16-Unet*	13.41	2.86	10.60	15.06	0.16s
FUSION-ROB*	13.90	3.14	11.04	15.69	2s
BMMNet*	14.37	5.10	10.92	15.51	0.1s
DABC [33]	14.49	4.08	12.72	15.53	0.7s
APMoE [27]	14.74	3.88	11.74	15.63	0.2s
CSWS [31]	14.85	3.48	11.84	16.38	0.2s
Ours single	14.58	3.96	11.50	15.24	0.1s
Ours cross-stich [41]	14.33	3.85	11.23	15.14	0.1s
Ours	13.08	2.72	10.27	13.95	0.2s

based on ResNet-50 and trained for the three tasks on NYUD-v2, and jointly depth prediction and semantic segmentation on SUN-RGBD. The performance on NYUD-v2 dataset is shown in Table 5. We can observe that the performances of our PAP method are superior or competitive, though using only RGB images as input. Such results can demonstrate that although depth ground truth is not directly use, our method can benefit the segmentation from jointly learning depth information. The performances on SUN-RGBD dataset are illustrated in Table 6, we can see that though slightly weaker than RDF-152 [48] in mean-acc metric, our method can obtain best results in other metrics. Such results reveal that our predictions are superior or at least competitive with state-of-the-art methods. Visualized results can be observed in Fig. 8, we can see that our predictions are with high quality and close to ground truth.

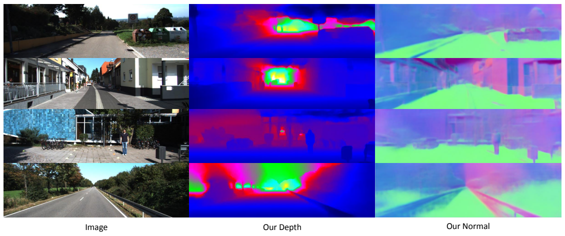


Figure 9. Qualitative results of our method on KITTI dataset. We can find that our model obtains good depth predictions and normal estimations.

5.5. Effectiveness On Distilling

Sometimes the ground truth data cannot be always available for each task, e.g., some widely-used outdoor depth datasets, such as KITTI [53], has no or very limited surface normal and segmentation ground truth. However, we can use PAP method to distill the knowledge from other dataset to boost the target task. We train our model on NYUD-v2 for depth and normal estimation, and then freeze the normal branch to train the model on KITTI. We evaluate our predictions on the KITTI online evaluation server, and the results are shown in Table 7 (* means anonymous method). Our PAP method outperforms our single-task and cross-stich based model. Compared with the state-of-the-art methods, though slightly weaker than DORN [15], our method obtains superior performances than all other published or unpublished approaches. Note that our method runs faster than DORN, which can be seen as a trade-off. These results demonstrate the effectiveness and potential of PAP method on task distilling and transferring. Qualitative results can be seen on Fig. 9, and our predictions on depth and normal are both with high quality.

6. Conclusion

In this paper, we propose a novel Pattern-affinitive Propagation method for jointly predicting depth, surface normal and semantic segmentation. Statistic results have shown that the affinitive patterns among tasks can be modeled in pair-wise similarities to some extent. The PAP can effectively learn the pair-wise relationships from each task, and further utilize such cross-task complementary affinity to boost and regularize the joint task learning procedure via the cross-task and task-specific propagation. Extensive experiments demonstrate our PAP method obtained state-of-the-art or competitive results on these three tasks.In the future, we may generalize and improve the efficiency of the method on more vision tasks.

7. Acknowledgement

This work was supported by the National Science Fund of China under Grant Nos. U1713208, 61806094, 61772276 and 61602244, Program for Changjiang Scholars, and 111 Program AH92005.

References

[1] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *CVPR*, pages 5965–5974, 2016.

[2] Gedas Bertasius, Lorenzo Torresani, X Yu Stella, and Jianbo Shi. Convolutional random walk networks for semantic image segmentation. In *CVPR*, pages 6137–6145, 2017.

[3] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[4] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, pages 2722–2730, 2015.

[5] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *ECCV*, pages 108–125, 2018.

[6] Yanhua Cheng, Rui Cai, Zhiwei Li, Xin Zhao, and Kaiqi Huang. Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation. In *CVPR*, volume 3, pages 1475–1483, 2017.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[8] Zhuo Deng, Sinisa Todorovic, and Longin Jan Latecki. Semantic segmentation of rgbd images with mutex constraints. In *ICCV*, pages 1733–1741, 2015.

[9] L Di, Chen Guangyong, Cohen-Or Daniel, Heng Pheng-Ann, and Huang Hui. Cascaded feature network for semantic segmentation of rgb-d images. In *ICCV*, pages 1320–1328, 2017.

[10] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, pages 2650–2658, 2015.

[11] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, pages 2366–2374, 2014.

[12] Terrence Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3-4):143–166, 2003.

[13] David F Fouhey, Abhinav Gupta, and Martial Hebert. Data-driven 3d primitives for single image understanding. In *ICCV*, pages 3392–3399, 2013.

[14] David Ford Fouhey, Abhinav Gupta, and Martial Hebert. Unfolding an indoor origami world. In *ECCV*, pages 687–702, 2014.

[15] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pages 2002–2011, 2018.

[16] Ross Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448, 2015.

[17] Saurabh Gupta, Ross Girshick, Pablo Arbelez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, volume 8695, pages 345–360, 2014.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollr, and Ross Girshick. Mask R-CNN. *ICCV*.

[19] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, (6):1397–1409, 2013.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[21] Yang He, Wei-Chen Chiu, Margret Keuper, Mario Fritz, and SI Campus. Std2p: Rgbd semantic segmentation using spatio-temporal data-driven pooling. In *CVPR*, pages 7158–7167, 2017.

[22] Yang He, Wei-Chen Chiu, Margret Keuper, Mario Fritz, and SI Campus. Std2p: Rgbd semantic segmentation using spatio-temporal data-driven pooling. pages 7158–7167, 2017.

[23] Tsung-Wei Ke, Jyh-Jing Hwang, Ziwei Liu, and Stella X. Yu. Adaptive affinity fields for semantic segmentation. In *ECCV*, 2018.

[24] Alex Kendall, Vijay Badrinarayanan, , and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[25] Seungryong Kim, Kihong Park, Kwanghoon Sohn, and Stephen Lin. Unified depth prediction and intrinsic image decomposition from a single image via joint convolutional neural fields. In *ECCV*, pages 143–159, 2016.

[26] Iasonas Kokkinos. Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, pages 5454–5463, 2017.

[27] Shu Kong and Charless Fowlkes. Pixel-wise attentional gating for parsimonious pixel labeling. *arXiv preprint arXiv:1805.01556*, 2018.

[28] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, pages 109–117, 2011.

[29] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *International Conference on 3D Vison*, pages 239–248, 2016.

[30] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):228–242, 2008.

[31] Bo Li, Yuchao Dai, and Mingyi He. Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *Pattern Recognition*, 2018.

[32] Bo Li, Chunhua Shen, Yuchao Dai, Anton van den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *CVPR*, pages 1119–1127, 2015.

[33] Ruibo Li, Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, and Lingxiao Hang. Deep attention-based classification network for robust depth prediction. *arXiv preprint arXiv:1807.03959*, 2018.

[34] Zhen Li, Yukang Gan, Xiaodan Liang, Yizhou Yu, Hui Cheng, and Liang Lin. Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In *ECCV*, pages 541–557, 2016.

[35] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, volume 1, pages 5168–5177, 2017.

[36] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, pages 3194–3203, 2016.

[37] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2016.

[38] Risheng Liu, Guangyu Zhong, Junjie Cao, Zhouchen Lin, Shiguang Shan, and Zhongxuan Luo. Learning to diffuse: A new perspective to design pdes for visual analysis. *IEEE transactions on pattern analysis and machine intelligence*, 38(12):2457–2471, 2016.

[39] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *NIPS*, pages 1520–1530, 2017.

[40] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 2017.

[41] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, pages 3994–4003, 2016.

[42] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R. Venkatesh Babu. Adadepth: Unsupervised content congruent adaptation for depth estimation. In *CVPR*, 2018.

[43] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, pages 1520–1528, 2015.

[44] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[45] Xiaojuan Qi, Renjie Liao, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. 3d graph neural networks for rgbd semantic segmentation. In *CVPR*, pages 5199–5208, 2017.

[46] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *CVPR*, pages 283–291, 2018.

[47] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, pages 5506–5514, 2016.

[48] Park Seong-Jin, Hong Ki-Sang, and Lee Seungyong. Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation. In *ICCV*, pages 4990–4999, 2017.

[49] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, pages 746–760, 2012.

[50] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[51] S. Song, S. P. Lichtenberg, and J. Xiao. Sun RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, pages 567–576, 2015.

[52] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *CVPR*, volume 2, 2017.

[53] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision*, pages 11–20, 2017.

[54] Peng Wang, Xiaohui Shen, Zhe Lin, and Scott Cohen. Towards unified depth and semantic prediction from a single image. In *CVPR*, pages 2800–2809, 2015.

[55] Peng Wang, Xiaohui Shen, Bryan Russell, Scott Cohen, Brian Price, and Alan L Yuille. Surge: Surface regularized geometry estimation from a single image. In *NIPS*, pages 172–180, 2016.

[56] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *CVPR*, pages 539–547, 2015.

[57] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, volume 10, 2018.

[58] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, 2018.

[59] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation. In *CVPR*, volume 1, pages 161–169, 2017.

[60] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*, pages 3917–3925, 2018.

[61] Bernhard Zeisl, Marc Pollefeys, et al. Discriminatively trained dense surface normal estimation. In *ECCV*, pages 468–484, 2014.

[62] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *ECCV*, pages 235–251, 2018.

[63] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017.