Monocular Depth Estimation with Affinity, Vertical Pooling, and Label Enhancement

Yukang Gan[1,2⋆], Xiangyu Xu[1,3⋆], Wenxiu Sun[1], and Liang Lin[1,2]

[1]SenseTime [2]Sun Yat-sen University [3]Tsinghua University
{ganyukang,xuxiangyu,sunwenxiu}@sensetime.com
linliang@ieee.org

Abstract. Significant progress has been made in monocular depth estimation with Convolutional Neural Networks (CNNs). While absolute features, such as edges and textures, could be effectively extracted, the depth constraint of neighboring pixels, namely relative features, has been mostly ignored by recent CNN-based methods. To overcome this limitation, we explicitly model the relationships of different image locations with an affinity layer and combine absolute and relative features in an end-to-end network. In addition, we consider prior knowledge that major depth changes lie in the vertical direction, and thus, it is beneficial to capture long-range vertical features for refined depth estimation. In the proposed algorithm we introduce vertical pooling to aggregate image features vertically to improve the depth accuracy. Furthermore, since the Lidar depth ground truth is quite sparse, we enhance the depth labels by generating high-quality dense depth maps with off-the-shelf stereo matching method taking left-right image pairs as input. We also integrate multi-scale structure in our network to obtain global understanding of the image depth and exploit residual learning to help depth refinement. We demonstrate that the proposed algorithm performs favorably against state-of-the-art methods both qualitatively and quantitatively on the KITTI driving dataset.

Keywords: monocular depth; affinity; vertical aggregation

1 Introduction

Depth estimation from images is a basic problem in computer vision, which has been widely applied in robotics, self-driving cars, scene understanding and 3D reconstruction. However, most works on 3D vision focus on the scenes with multiple observations, such as multiple viewpoints [22] and image sequences from videos [14], which are not always accessible in real cases. Therefore, monocular depth estimation has become a natural choice to overcome this problem, and substantial improvement has been made in this area with the rapid development of deep learning in recent years.

Specifically, most of the state-of-the-art methods [7, 12, 16] rely on Convolutional Neural Networks (CNNs) which learn a group of convolution kernels to

⋆ These two authors contribute equally to this study.

extract local features for monocular depth estimation. The learned depth feature for each pixel is calculated within the receptive filed of the network. It is an absolute cue for depth inference which represents the appearance of the image patch centered at the pixel, such as edges and textures. While these absolute features for each image location from convolution layer are quite effective in existing algorithms, it ignores the depth constraint between neighboring pixels.

Intuitively, neighboring image locations with similar appearances should have close depth, while the ones with different appearances are more likely to have quite large depth changes. Therefore, the relationship between different pixels, namely affinities, are very important features for depth estimation which have been mostly ignored by deep learning-based monocular depth algorithms. These affinities are different with the absolute features which are directly extracted with convolution operations. They are relative features which describes the similarities between the appearances of different image locations. And explicitly considering these relative features could potentially help the depth map inference.

In fact, affinities have been widely used in image processing methods, such as bilateral filter [25] which takes the spatial distance and color intensity difference as relative feature for edge-preserving filtering. More related to our work, affinities have also been used to estimate depth in a Conditional Random Field (CRF) framework [23], where the relative depth features are modeled as the differences between the gradient histograms computed from two neighboring patches. And the aforementioned depth constraint of neighboring pixels is enforced by the pairwise potential in the CRF.

Different with these methods, we learn to extract the relative features in neural network by introducing a simple yet effective affinity layer. In this layer, we define the affinity between a pair of pixels as the correlation of their absolute features. Thus, the relative feature from the affinity layer for one pixel is a vector composed of the correlation values with its surrounding pixels. By integrating the affinity layer into CNNs, we can seamlessly combine learned absolute and relative features for depth estimation in a fully end-to-end model. Since only the relationship between nearby pixels is important for depth inference, the proposed operation is conducted within a local region. In the proposed method, we only use the affinity operation at the lowest feature scale to reduce computational load.

Except for the constraint between neighboring pixels, we also consider another important observation in depth estimation that there are more depth changes in the vertical direction than in the horizontal [3]. In other words, objects tend to get further from the bottom to the top in many images. For example, in driving scenes, a road stretching vertically ahead in the picture often gets further away from the camera. Thus, to capture the local information in the vertical direction could potentially help refined depth estimation which motivates us to integrate vertical feature pooling in the proposed neural network.

To further improve the depth estimation results, we enhance the sparse depth ground truth from Lidar by exploiting the left-right image pairs. Different from previous methods which use photometric loss [9, 16] to learn disparities which are

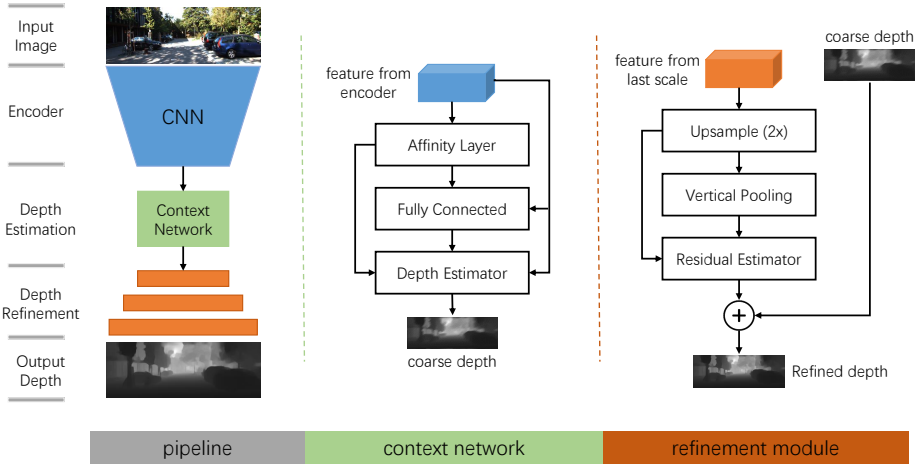

Fig. 1. An overview of the proposed network. The network is composed of a deep CNN for encoding image input, a context network for estimating coarse depth, and a multi-scale refinement module to predict more accurate depth. The context network adopts affinity and fully-connected layers to capture neighboring and global context information, respectively. The refinement module upsamples the coarse depth gradually by learning residual maps with features from previous scale and vertical pooling.

inversely proportional to image depth, we adopt an off-the-shelf stereo matching method to predict dense depth from the image pairs and then use the predicted high-quality dense results as auxiliary labels to assist the training process.

We conduct comprehensive evaluations on the KITTI driving dataset and show that the proposed algorithm performs favorably against state-of-the-art methods both qualitatively and quantitatively. Our contributions could be summarized as follows.

- We propose a neighboring affinity layer to extract relative features for depth estimation.
- We propose to use vertical pooling to aggregate local feature to capture long-range vertical information.
- We use stereo matching network to generate high-quality depth predictions from left-right image pairs to assist the sparse Lidar depth ground truth.
- In addition, we adopt a multi-scale architecture to obtain global context and learn residual maps for better depth estimation.

2 Related Work

2.1 Supervised Depth Estimation.

Supervised approaches take one single RBG image as input and use measured depth maps from RGB-D cameras or laser scanners as ground-truth for training.

Saxena *et al.* [23] propose a learning-based approach to predict the depth map as a function of the input image. They adopt Markov Random Field(MRF) that incorporates multi-scale hand-crafted texture features to model both depths at individual points as well as the relation between depths at different points. [23] is later extended to a patch-based model known as Make3D [24] which first uses MRF to predict plane parameters of the over-segmented patches and then estimates the 3D location and orientation of these planes. We also model the relation between depths at different points. But instead of relying on hand-crafted features, we integrate a correlation operation into deep neural networks to obtain more robust and general representation.

Deep learning achieves promising results on many applications [12, 3, 28, 29]. Many recent works [7, 6, 27] utilize the powerful Convolutional Neural Networks(CNN) to learn image features for monocular depth estimation. Eigen *et al.* [7, 6] employ multi-scale deep network to predict depth from single image. They first predict a coarse global depth map based on the entire image and then refine the coarse prediction using a stacked neural network. In this paper, we also adopt multi-scale strategy to perform depth estimation. But we only predict depth map at the coarsest level and learn to predict residuals afterwards which helps refine the estimation. Li *et al.* [18] also use a DCNN model to learn the mapping from image patches to depth values at the super-pixel level. A hierarchical CRF is then used to refine the estimated super-pixel depth to the pixel level. Furthermore, there are several supervised approaches that adopt different techniques such as depth transfer from example images [15, 21], incorporating semantic information [20, 17], and formulating depth estimation as pixel-wise classification task [2].

2.2 Unsupervised Depth Estimation

Recently, several works attempt to train monocular depth prediction model in an unsupervised way which does not require ground truth depth at training time. Garg *et al.* [9] propose an encoder-decoder architecture which is trained for single image depth estimation on an image alignment loss. This method only requires a pair of images, source and target, at training time. To obtain the image alignment loss, the target image is warped to reconstruct the source image using the predicted depth. Godard *et al.* [12] extend [9] by enforcing consistency between the disparities produced relative to both the left and right images. Besides image reconstruction loss, this method also adopts appearance matching loss, disparity smooth loss and left-right consistency loss to produce more accurate disparity maps. Xie *et al.* [26] propose a novel approach which tries to synthesized the right view when given the left view. Instead of directly regressing disparity values, they produce probability maps for different disparity level. A selection layer is then utilized to render the right view using these probability maps and the given left view. The whole pipeline is also trained on a image reconstruction loss. Unlike the above methods that are trained using stereo images, Zhou *et al.* [30] propose to train an unsupervised learning framework on unstructured video sequences. They adopt a depth CNN and a pose CNN to

estimate monocular depth and camera motion simultaneously. The nearby views are warped to the target view using the computed depth and pose to calculate the image alignment loss. Instead of using view synthesis as the supervisory signal, we employ a powerful stereo matching approach [22] to predict dense depth map from the stereo images. The predicted dense depth map, together with the sparse velodyne data, are used as ground truth during our training.

2.3 Semi-/Weakly Supervised Depth Estimation

Only few works fall in the line of research in semi- and weakly supervised training of single image depth prediction. Chen *et al.* [3] present a new approach that learns to predict depth map in unconstrained scenes using annotations of relative depth. But the annotations of relative depth only provides indirect information on continuous depth values. More recently, Kuznietsov *et al.* [16] propose to train a semi-supervise model using both sparse ground truth and unsupervised cues. They use ground truth measurement to solve the ambiguity of unsupervised cues and thus do not require coarse-to-fine image alignment loss during training.

2.4 Feature Correlations

Other works have attempted to explore correlations in feature maps in the context of classification [19, 8, 5]. Lin et al. [19] utilize bilinear CNNs to model local pairwise feature interactions . While the final representation of a full bilinear pooling is very high-dimensional, Gao et al. [8] reduce the feature dimensionality via two compact bilinear pooling. In order to capture higher order interactions of features, Cui et al. [5] proposed a kernel pooling scheme and combine it with CNNs. Instead of adopting bilinear models to obtain discriminative features, we propose to model feature relationships between neighboring image patches to provide more information for depth inference.

3 Method

An overview of our framework is shown in Figure 1. The proposed network adopts an encoder-decoder architecture, where the input image is first transformed and encoded as absolute feature maps by a deep CNN feature extractor. Then a context network is used to capture both neighboring and global context information with the absolute features. Specifically, we propose an affinity layer to model relative features within a local region of each pixel. By combining the absolute and relative features with a fully-connected layer, we obtain global features which indicates the global layout and properties of the image. The global features of the fully-connected layer, the absolute features from the deep encoder, and the relative features are fed into our depth estimator, a multi-layer CNN, to generate an initial coarse estimate of the image depth. In the meanwhile, we also take these features as initial input of the following multi-scale refinement modules. The refinement network at each scale is composed of a proposed vertical pooling

Fig. 2. Examples of the enhanced dense depth maps generated by a stereo matching model [22]. We use these depth maps as complementary data to the sparse ground truth depth maps. The left column contains RGB images, while the middle and right column show the enhanced depth maps and sparse ground truth, respectively.

layer which aggregates local depth information vertically, and a residual estimator which learns residual map for refining the coarse depth estimation from the last scale. Both the features from previous scale and the proposed vertical pooling layer are used in the residual estimator.

3.1 Affinity Layer

While the relationships between neighboring pixels, namely affinities, are very important cues for inferring depth, they cannot be explicitly represented in a vanilla CNN model. To overcome this limitation, we propose an affinity layer to learn these cues and combine absolute and relative features for superior depth estimation.

For concise and effective formulation, we define the affinity as the correlation between the absolute features of two image pixels. Since the absolute features represents the local appearance of image locations, such as edges and textures, the correlation operation could effectively model the appearance similarities between these pixels. Mathematically, this operation could be formulated as:

$$\mathbf{v}(\mathbf{x})_{m,n} = \mathbf{f}(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x} + (m, n)]);\ m, n \in [-k, k] \tag{1}$$

where $\mathbf{v}(\mathbf{x}) \in R^{(2k+1)\times(2k+1)}$ represents the affinities of location $\mathbf{x}$ calculated in a squared local region of size $(2k+1) \times (2k+1)$. $\mathbf{f}(\mathbf{x})$ is the absolute feature vector from the convolutional feature extractor layer at location $\mathbf{x}$. In fact, we can reshape $\mathbf{v}(\mathbf{x})$ into a 1-dimensional vector of size $1 \times (2k+1)^2$, and the relative features of a input image become $(2k+1)^2$ feature maps which could be fed into the following estimation and refinement layers. Suppose the input feature map is of size $w \times h \times c$ where w, h and c are the width, height and channels, respectively. $w \times h \times c \times (2k+1)^2$ multiplications are needed for computing the relative feature which is computationally heavy. To remedy the problem of the square complexity

of the affinity operation, we only perform this layer on the lowest feature scale (in the context network in Figure 1) to reduce the computational load. The proposed affinity layer is integrated in the CNN model and works complementarily with the absolute features, which significantly helps depth estimation.

3.2 Task Specific Vertical Pooling

Depth distribution in real world scenarios has a special kind of pattern that the majority of depth changes lies in the vertical direction. *e.g.* The road often stretches to the far side alone the vertical direction. The faraway objects, such as sky and mountains, are more likely to be located at the top of a landscape picture. Recognizing this kind of patterns can provide useful information for accurate single image depth estimation. However, due to the lack of supervision and huge parameters space, normal operations in deep neural network such as convolution and pooling with squared filters may not be effective in finding such patterns. Furthermore, a relative large squared pooling layer aggregates too much unnecessary information from horizontal locations while it is more efficient to consider vertical features only.

In this paper, we propose to obtain the local context in vertical direction through vertical pooling layer. The vertical pooling layer uses average pooling with kernels of size $H \times 1$ and outputs feature maps of equal size with the input features. Multiple vertical pooling layers with different kernel heights are used in our network to handle feature maps across different scales. Specifically, we use four kernels of size 5×1, 7×1, 11×1 and 11×1 to process feature maps of scale $S/8$, $S/4$, $S/2$ and S, where S denotes the resolution of input images. More detailed analysis of vertically aggregating depth information are presented in Section 4.5.

3.3 Multi-Scale Learning

As shown in Figure 1, our model predicts a coarse depth map through a context network. Besides exploiting local context using operations mentioned in the preview sections, we follow [7] to take advantage of fully connected layers to integrate a global understanding of the full scene into our network. The output feature maps of the encoder and the self-correlation layer are taken as input of the fully connected layer. The output feature vector of fully connected layer is then reshaped to produce the final output feature map which is at the 1/8-resolution compared to the input image.

Given the coarse depth map, our model learns to refine the coarse depth by adopting the residual learning scheme propose by He *et al.* [13]. The refinement module first up-sample the input feature map by factor of 2. A residual estimator then learns to predict the corresponding residual signal based on the up-sampled feature, the local context feature and the long skip connected low level feature. Without the need to predict absolute depth values, the refinement module can focus on learning residual that helps produce accurate depth maps. Such learning strategy can lead to smaller network and better convergence. Several refinement

modules are employed in our model to produce residuals across multiple scales. The refinement process can be formulated as:

$$d_s = UP\{d_{s+1}\} + r_s \qquad 0 \leq s \leq S \tag{2}$$

where d_s and r_s denote depth and residual maps that are downsampled by a factor of 2^s from full resolution size. $UP\{\cdot\}$ denotes $2\times$ upsample operation. We supervise the estimated depth map across $S+1$ scales. Ablation study in Section 4.5 demonstrates that incorporating residual learning can lead to more accurate depth maps compared to direct learning strategy.

3.4 Loss Function

Ground truth enhancement. The ground truth depth maps obtained from Lidar sensor are too sparse (only 5% pixels are valid) to provide enough supervisory signal for training a deep model. In order to produce high quality, dense depth maps, we enhance the sparse ground truth with dense depth maps predicted by a stereo matching approach [22]. We use both the dense depth maps and the sparse velodyne data as ground truth at training time. Some samples of predicted depth maps are shown in Fig 2.

Training loss. The enhanced dense depth maps produced by stereo matching model are not accurate enough compared to ground truth depth maps. The error between predicted and ground truth depth maps is shown in Table **??**. We use a weighted sum L2 loss to suppress the noise contained in the enhanced dense depth maps:

$$Loss = \sum_{i \in \Lambda} \|pred_i - gt_i\|_2^2 + \alpha * \sum_{i \in \Omega} \|pred_i - gt_i\|_2^2 \tag{3}$$

where $pred_i$ and gt_i denote the predicted depth and ground truth depth at i_{th} pixel. Λ denotes a collection of pixels where sparse ground truth values are valid. Ω denotes a collection of pixels where sparse ground truth values are invalid and values from enhance depth maps are used as ground truth. α is set to 0.3 in all the experiments.

4 Experiments

We show the main results in this section and present more evaluations in the supplementary material.

4.1 Dataset

We evaluate our approach on the publicly available KITTI dataset [10], which is a widely-used dataset in the field of single image depth estimation. The dataset contains over 93 thousand semi-dense depth maps with corresponding Lidar scans and RGB images. All the images in this dataset are taken from a driving

Table 1. Quantitative results of our method and approaches reported in the literature on the test set of the KITTI Raw dataset used by Eigen *et al.* [7] for different caps on ground-truth and/or predicted depth. Enhanced depth denotes the depth maps generated by [22]. Best results shown in bold.

		ARD	SRD	RMSE	RMSE(log)	δ <1.25	δ <1.25^2	δ <1.25^3
Approach	cap	lower is better				higher is better		
Eigen *et al.* [7]	0 - 80 m	0.215	1.515	7.156	0.270	0.692	0.899	0.967
Liu *et al.* [21]	0 - 80 m	0.217	1.841	6.986	0.289	0.647	0.882	0.961
Zhou *et al.* [30]	0 - 80 m	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Godard *et al.* [12]	0 - 80 m	0.114	0.898	4.935	0.206	0.861	0.949	0.976
Kuznietsov *et al.* [16]	0 - 80 m	0.113	0.741	4.621	0.189	0.862	0.960	**0.986**
Ours	0 - 80 m	**0.098**	**0.666**	**3.933**	**0.173**	**0.890**	**0.964**	0.985
Enhanced depth	0 - 80 m	0.025	0.075	1.723	0.049	0.994	0.998	0.999
Zhou *et al.* [30]	1 - 50 m	0.190	1.436	4.975	0.258	0.735	0.915	0.968
Garg *et al.* [9]	1 - 50 m	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Godard *et al.* [12]	1 - 50 m	0.108	0.657	3.729	0.194	0.873	0.954	0.979
Kuznietsov *et al.* [16]	1 - 50 m	0.108	0.595	3.518	0.179	0.875	0.964	**0.988**
Ours	1 - 50 m	**0.094**	**0.552**	**3.133**	**0.165**	**0.898**	**0.967**	0.986

car in an urban scenario, with a typical image resolution being 1242×375. In order to perform fair comparisons with existing work, we adopt the split scheme proposed by Eigen *et al.* [7] which splits the total 56 scenes from raw KITTI dataset into 28 for training and 28 for testing. Specifically, we use 22,600 images for training and the rest for validation. The evaluation is performed on the test split of 697 images. We also adopt the KITTI split provided by KITTI stereo 2015, which provides 200 high quality disparity images from 28 scenes. We use the 30,159 images from the remaining scenes as training set. While the 200 disparity images provides more depth information than the sparse, reprojected velodyne laser data, they have CAD modes inserted in place of moving cars. We evaluate our model on these high quality disparity images to obtain more convincing demonstrations.

4.2 Evaluation Metrics

We evaluate the performance of our approach in monocular depth prediction using the velodyne ground truth data on the test images. We follow the depth evaluation metrics used by Eigen *et al.* [7]:

ARD: $\frac{1}{|T|}\sum_{y\in T}|y-y^*|/y^*$ **RMSE:** $\sqrt{\frac{1}{|T|}\sum_{y\in T}\|y-y^*\|^2}$

SRD: $\frac{1}{|T|}\sum_{y\in T}\|y-y^*\|^2/y^*$ **RMSE(log):** $\sqrt{\frac{1}{|T|}\sum_{y\in T}\|logy-logy^*\|^2}$

Threshold: % of y_i s.t. $max(\frac{y_i}{y^*}, \frac{y^*}{y_i}) = \delta < thr$

where T denotes a collection of pixels where the ground truth values are valid. y^* denotes the ground truth value.

Table 2. Quantitative results of different variants of our method on the test set of the KITTI Raw dataset used by Eigen *et al.* [7] without capping the ground-truth. Baseline† denotes the baseline model that is trained using velodyne data and stereo images. Baseline‡ denotes the baseline model that is trained using velodyne data and predicted dense depth maps. Ours§ denotes a variant of our model which utilizes squared average pooling layers. Ours¶ denotes a variant of our model which utilizes horizontal pooling layers. Legend: **R:** only predict depth map at the coarsest level and learn to predict residual for refinement afterwards. **A:** include affinity learning operation. **V:** use vertical pooling layer to obtain task specific context. **G:** include global context.

					ARD	SRD	RMSE	RMSE(log)	δ <1.25	δ <1.25^2	δ <1.25^3
Method	R	A	V	G	lower is better				higher is better		
Baseline†					0.120	0.757	4.734	0.202	0.856	0.953	0.972
Baseline‡					0.117	0.748	4.620	0.191	0.861	0.958	0.978
Ours	✓				0.115	0.740	4.514	0.189	0.865	0.958	0.980
	✓	✓			0.106	0.696	4.231	0.178	0.882	0.960	0.982
Ours¶	✓	✓			0.104	0.694	4.141	0.179	0.882	0.961	0.982
Ours§	✓	✓			0.102	0.683	4.132	0.177	0.884	0.962	0.982
Ours	✓	✓	✓		0.102	0.674	4.027	0.174	0.889	0.962	0.982
	✓	✓	✓	✓	**0.098**	**0.667**	**3.934**	**0.173**	**0.890**	**0.963**	**0.984**

4.3 Implementation Details

We implement our method using the publicly available TensorFlow [1] framework. The whole model is an hour-glass structure in which Resnet50 is utilized as the encoder. We trained our model from scratch for 80 epochs, with a batch size of 8 using the Adam method with $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The learning rate is initialized as 10^{-4} and exponentially decayed by 10 every 30 epochs during training. All the parameters in our model are initialized based on xavier algorithm [11]. It costs about 7G of GPU memory and 50 hours to train our model on a single NVIDIA GeForce GTX TITAN X GPU with 12GB memory. The average training time for each image is less than 100 ms and it takes less than 70 ms to test one image.

Data augmentation is also conducted during training process. The input image is flipped with a probability of 0.5. We randomly crop the original image into size of $2h \times h$ to retain image ratio, where h is the height of the original image. The input image is obtained by resizing the cropped image to a resolution of 512×256. We also performed random brightness for color augmentation, with 50% chance, by sampling from a uniform distribution in the range of [0.5, 2.0].

4.4 Comparisons with state-of-the-art methods

Table 1 shows the quantitative comparisons between our model and other state-of-the-art methods in monocular depth estimation. It can be observed that our method achieves best performances for all evaluation metrics at both 80m and

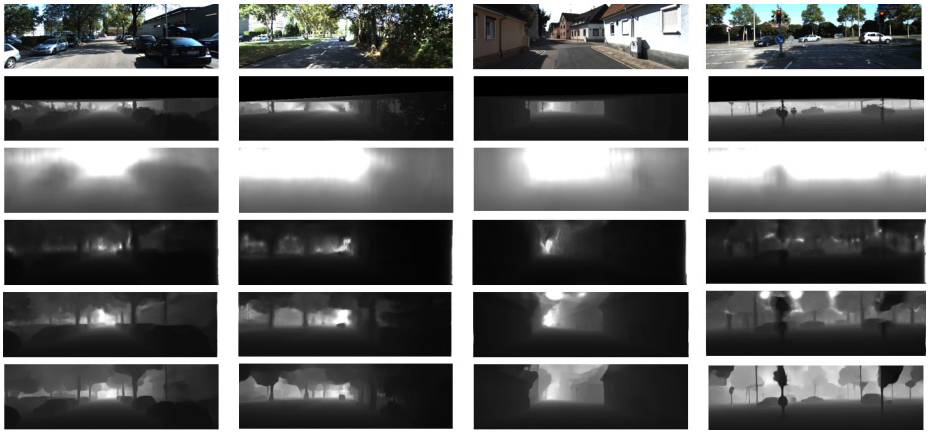

Fig. 3. Qualitative results on the test set of the KITTI Raw dataset used by Eigen *et al.* [7]. From top to bottom, the images are input, ground truth, results of Eigen *et al.* [7], results of Garg *et al.* [9], results of Godard *et al.* [12] and results of our method, respectively. Sparse ground truth have been interpolated for better visualization.

50m caps, except for the accuracy at δ <1.25^3 where we obtain comparable results with Kuznietsov *et al.* [16] at cap of 80m (0.985 *vs* 0.986) and 50m (0.986 *vs* 0.988). Specifically, our method reduces the RMSE metric by 20.3% compared with Godard *et al.* [12] and 14.9% compared with Kuznietsov *et al.* [16] at the cap of 80 m. Furthermore, our model obtain accuracy of 89.0% and 89.8% at δ <1.25^2 metric at the cap of 80 m and 50 m, outperforming Kuznietsov *et al.* [16] by 2.8% and 2.4% respectively.

To further evaluate the performance of our approach, we train a variant model on the training set of the official KITTI split and perform evaluation on the KITTI 2015 stereo training set which contains 200 high quality disparity images. We convert these disparity images into depth maps for evaluation using the camera parameters provided by KITTI dataset. The result is shown in Table 3. It can be observed that our method outperforms [12] by a large margin and achieves close results with the variant model of Godard *et al.* [12] which is trained and tested with two input images.

We provide qualitative comparisons in Figure 3 which shows that our results are visually more accurate than the compared methods. Some qualitative results on Cityscape dataset [4] and Make3D dataset [24] are shown in Figure 5, which are estimated by our model that is trained only on KITTI dataset. The high quality results show that our model can generalize well on unseen scenes. The comparisons performed above well demonstrate the superiority of our approach in predicting accurate depth map from single images. More qualitative results on KITTI dataset are shown in Figure 4.

Table 3. Comparisons of our method and two different approaches. Results on the KITTI 2015 stereo 200 training set images [10]. Best results shown in bold.

	ARD	SRD	RMSE	RMSE(log)	δ <1.25	δ <1.25^2	δ <1.25^3
Approach	lower is better				higher is better		
[12] with Deep3Ds [26]	0.151	1.312	6.344	0.239	0.781	0.931	0.976
Godard *et al.* [12]	0.097	0.896	5.093	0.176	0.879	0.962	0.986
Ours	**0.079**	**0.500**	**3.522**	**0.137**	**0.918**	**0.978**	**0.989**
Godard *et al.* [12] stereo	0.068	0.835	4.392	0.146	0.942	0.978	0.989

4.5 Ablation study

In this subsection, we show effectiveness and necessity of each component in our proposed model and also demonstrate the effectiveness of the network design.

Supervisory signal: To validate the effectiveness of using predicted dense depth maps as ground truth at training time. We compare our baseline model (denoted as Baseline‡) with a variant (denoted as Baseline†) which is trained using image alignment loss. Results are shown in the first two rows of Table 2. It can be easily observed that Baseline‡ achieves better results than Baseline† on all the metrics. This may due to the well known fact that stereo depth reconstruction based on image matching is an ill-pose problem. Training on a image alignment loss may provide inaccurate supervisory signal. On the contrary, the dense depth maps used in our method are more accurate and more robust against the ambiguity, since they are produced by a powerful stereo matching model [22] which is well designed and trained on massive data for the task of depth reconstruction. Thus, the superior result, together with the above analysis, well validate that utilizing predicted depth maps as ground truth can provide more useful supervisory signal.

Residual learning *vs* direct learning: The baseline model of our approach (denoted as Baseline‡) is implemented using direct learning strategy which learns to output the depth map directly instead of the residual depth map. Note that the baseline model represents our network without any of the components R, A, V, G in Table 2. As shown in Table 2, the baseline model achieves 0.117 at ARD metric and 4.620 at RMSE metric. In order to compare residual learning strategy with direct learning strategy, we replace direct learning with residual learning in Baseline‡ and keep other settings identical to obtain a variant model with residual learning strategy. The performance of this variant model is shown in the third row of Table 2, which outperforms Baseline‡ with slight improvements on all the metrics. This may due the reason that residual learning can focus on modeling the highly non-linear residuals while direct learning needs to predict absolute depth values. Moreover, residual learning also helps alleviate the problem of over-fitting [13].

Table 4. Quantitative results on NYU Depth v2 dataset(part). H-pooling denotes horizontal pooling. Note that our model was trained on the labeled training set with 795 images instead of the full dataset which contains 20K images.

Method	δ <1.25	δ <1.25^2	δ <1.25^3	rel	log_{10}	rms
w/ H-pooling	0.747	0.929	0.977	0.165	0.069	0.652
w/o affinity	0.732	0.920	0.972	0.179	0.075	0.694
Ours	**0.756**	**0.934**	**0.980**	**0.158**	**0.066**	**0.631**

Fig. 4. More qualitative results on KITTI test splits.

Pooling methods: To validate the idea that incorporating local context through pooling layers helps boost the performance of depth estimation, we implement three variant models that use vertical pooling layers, horizontal pooling layers (denoted as Ours¶) and squared average pooling layers (denoted as Ours§). Note that we also use multiple average pooling layers with kernels of different sizes to handle multi-scale feature maps. Specifically, we use four squared average pooling layers in Ours§ whose kernel sizes are set to 5×5, 7×7, 11×11 and 11×11 respectively. The results are shown in the middle three lines of Table 2. As one can see, by adopting squared average pooling layers, the model achieves slightly better results where SRD metric is reduced from 0.696 to 0.683 while RMSE metric is reduced from 4.231 to 4.132. The improvement demonstrates the effectiveness of exploiting local context through pooling layers. Similar improvements can be observed by integrating horizontal pooling layers. Furthermore, by replacing squared average polling layers with vertical pooling layers, our model obtains better results with more significant improvements where SRD metric is reduced from 0.696 to 0.674 while RMSE metric is reduced from 4.231 to 4.027. The further improvement proves that vertical pooling is able to model the local context more effectively compared to squared average pooling and horizontal pooling. This may due to the reason that squared average pooling combines both the depth distribution alone the horizontal and vertical direction which might introduce noise and redundant information.

Contribution of each component: To discover the vital elements in our proposed method, we conduct ablation study by gradually integrating each component into our model. The results are shown in Table 2. Besides the improvements brought by residual learning and vertical pooling modules which have been analyzed in the above comparisons, integrating affinity layer can result in major

Fig. 5. Qualitative results on Make3D dataset [24] (left two columns) and Cityscape dataset [4] (right two columns).

improvements on all the metrics. More specifically, ARD metric is reduced from 0.115 to 0.106, RMSE metric is reduced from 4.514 to 4.231 and the accuracy at δ <1.25 is boosted from 0.865 to 0.882. This proves that affinity layer is the key component of our proposed approach and thus well validate the insight that explicitly considering relative features between neighboring patches can help the monocular depth estimation. Moreover, integrating fully connected layers to exploit global context information further boosts the performance of our model. It can be seen from the last row of Table 2 that accuracy at δ <1.25^3 is further improved to 0.984. This shows that some challenge outliers can be predicted more accurately given the global context information.

We conduct more experiments to evaluate the proposed components on the NYUv2 dataset in Table 4. The *rel* error increases from 0.158 to 0.165 if we replace vertical pooling layer with horizontal pooling layer. Similar performance drop can be observed after we remove the affinity layer where the *rms* error increases from 0.631 to 0.694. The Results further prove that affinity layer and vertical pooling both play an important role in improving the estimation performance, which also shows that proposed method generalizes well to the NYUv2 dataset.

5 Conclusions

In this work, we propose a novel affinity layer to model the relationship between neighboring pixels, and integrate this layer into CNN to combine absolute and relative features for depth estimation. In addition, we exploit the prior knowledge that vertical information potentially helps depth inference and develop vertical pooling to aggregate local features. Furthermore, we enhance the original sparse depth labels by using stereo matching network to generate high-quality depth predictions from left-right image pairs to assist the training process. We also adopt a multi-scale architecture with residual learning for improved depth estimation. The proposed method performs favorably against the state-of-the-art monocular depth algorithms both qualitatively and quantitatively. In future work, we will investigate more about the generalization abilities of the affinity layer and vertical pooling for indoor scenes. It will also be interesting to explore more detailed geometry relations and semantic segmentation information for more robust depth estimation.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: OSDI. vol. 16, pp. 265–283 (2016)
2. Cao, Y., Wu, Z., Shen, C.: Estimating depth from monocular images as classification using deep fully convolutional residual networks. IEEE Transactions on Circuits and Systems for Video Technology (2017)
3. Chen, W., Fu, Z., Yang, D., Deng, J.: Single-image depth perception in the wild. In: Advances in Neural Information Processing Systems. pp. 730–738 (2016)
4. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
5. Cui, Y., Zhou, F., Wang, J., Liu, X., Lin, Y., Belongie, S.J.: Kernel pooling for convolutional neural networks. In: CVPR. vol. 1, p. 7 (2017)
6. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2650–2658 (2015)
7. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in neural information processing systems. pp. 2366–2374 (2014)
8. Gao, Y., Beijbom, O., Zhang, N., Darrell, T.: Compact bilinear pooling. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 317–326 (2016)
9. Garg, R., BG, V.K., Carneiro, G., Reid, I.: Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vision. pp. 740–756. Springer (2016)
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 3354–3361. IEEE (2012)
11. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Aistats. vol. 9, pp. 249–256 (2010)
12. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR. vol. 2, p. 7 (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. Karsch, K., Liu, C., Kang, S.B.: Depth extraction from video using non-parametric sampling. In: European Conference on Computer Vision. pp. 775–788. Springer (2012)
15. Konrad, J., Wang, M., Ishwar, P.: 2d-to-3d image conversion by learning depth from examples. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on. pp. 16–22. IEEE (2012)
16. Kuznietsov, Y., Stückler, J., Leibe, B.: Semi-supervised deep learning for monocular depth map prediction. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6647–6655 (2017)
17. Ladicky, L., Shi, J., Pollefeys, M.: Pulling things out of perspective. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 89–96 (2014)

18. Li, B., Shen, C., Dai, Y., van den Hengel, A., He, M.: Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1119–1127 (2015)
19. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1449–1457 (2015)
20. Liu, B., Gould, S., Koller, D.: Single image depth estimation from predicted semantic labels. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 1253–1260. IEEE (2010)
21. Liu, M., Salzmann, M., He, X.: Discrete-continuous depth estimation from a single image. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. pp. 716–723. IEEE (2014)
22. Pang, J., Sun, W., Ren, J., Yang, C., Yan, Q.: Cascade residual learning: A two-stage convolutional neural network for stereo matching. In: International Conf. on Computer Vision-Workshop on Geometry Meets Deep Learning (ICCVW 2017). vol. 3 (2017)
23. Saxena, A., Chung, S.H., Ng, A.Y.: Learning depth from single monocular images. In: Advances in neural information processing systems. pp. 1161–1168 (2006)
24. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. IEEE transactions on pattern analysis and machine intelligence **31**(5), 824–840 (2009)
25. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Computer Vision, 1998. Sixth International Conference on. pp. 839–846. IEEE (1998)
26. Xie, J., Girshick, R., Farhadi, A.: Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In: European Conference on Computer Vision. pp. 842–857. Springer (2016)
27. Xu, D., Ricci, E., Ouyang, W., Wang, X., Sebe, N.: Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In: Proceedings of CVPR. vol. 1 (2017)
28. Xu, X., Pan, J., Zhang, Y.J., Yang, M.H.: Motion blur kernel estimation via deep learning. TIP **27**, 194–205 (2018)
29. Xu, X., Sun, D., Pan, J., Zhang, Y., Pfister, H., Yang, M.H.: Learning to super-resolve blurry face and text images. In: ICCV (2017)
30. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR. vol. 2, p. 7 (2017)