High-Precision Self-Supervised Monocular Depth Estimation with Rich-Resource Prior

Wencheng Han[®] and Jianbing Shen [†][®],

SKL-IOTSC, Computer and Information Science, University of Macau, China

Abstract. In the area of self-supervised monocular depth estimation, models that utilize rich-resource inputs, such as high-resolution and multiframe inputs, typically achieve better performance than models that use ordinary single image input. However, these rich-resource inputs may not always be available, limiting the applicability of these methods in general scenarios. In this paper, we propose Rich-resource Prior Depth estimator (RPrDepth), which only requires single input image during the inference phase but can still produce highly accurate depth estimations comparable to rich-resource based methods. Specifically, we treat rich-resource data as prior information and extract features from it as reference features in an offline manner. When estimating the depth for a single-image image, we search for similar pixels from the rich-resource features and use them as prior information to estimate the depth. Experimental results demonstrate that our model outperform other single-image model and can achieve comparable or even better performance than models with rich-resource inputs, only using low-resolution single-image input.

1 Introduction

Depth estimation is a crucial component in computer vision, particularly for applications like autonomous driving, where understanding the 3D structure of the environment is essential for navigation and decision-making. Traditionally, depth information has been obtained using stereo vision [14, 18] or LiDAR systems. However, these methods can be costly and complex, motivating the exploration of monocular depth estimation. Monocular depth estimation involves deducing the depth information of a scene from a single camera. This is inherently challenging as it requires the model to infer 3D information from 2D data, a task that humans do effortlessly but is complex for machines due to the loss of spatial information in a single image.

[†]Corresponding author: *Jianbing Shen*. This work was supported in part by the FDCT grants 0102/2023/RIA2, 0154/2022/A3, and 001/2024/SKL, the MYRG-CRG2022-00013-IOTSC-ICI grant and the SRG2022-00023-IOTSC grant.

Recent advancements in monocular depth estimation have opened avenues for simpler, more cost-effective solutions. Godard *et al.* [7] introduced a simplified self-supervised model for monocular depth estimation. They employ innovative loss functions and sampling methods to achieve promising depth accuracy. Subsequently, many other methods improve the performance further by designing better network architectures [11, 19], using more suitable loss functions [13, 22, 22, 31]. Watson *et al.* [32] proposed an adaptive deep end-to-end cost volume-based method for dense depth estimation. Their method utilizes sequence information at test time and introduces a novel consistency loss to enhance the performance of self-supervised monocular depth estimation networks. Although this method achieves a significant improvement compared to previous works, it requires richer-resource inputs, specifically multi-frame data, during inference. Many methods follow this approach and propose highly effective depth estimators using multi-frame data inputs [5, 10].

In this paper, we refer to high-resolution, multi-frame data as "rich-resource data". We have noticed that many of the best-performing methods depend on rich-resource data. This poses significant challenges in real-world scenarios. In some situations, acquiring rich-resource inputs is impractical. For instance, multi-frame based models necessitate the capture of multi-frame data from varied positions. However, when cars are stationary, obtaining images from different positions is not possible. Moreover, many multi-frame based models demonstrate improved performance when future frames are available, but these cannot be obtained in real-world applications. Hence, there is a need for a method that can generate a comparable depth map to a rich-resource based model using only Low-Resolution (LR) single-image inputs.

To address this issue, we introduce a new self-supervised method for monocular depth estimation. The proposed method leverages features extracted from rich-resource inputs as prior information, allowing the accurate depth estimation using only LR single-image inputs during inference, as shown in Fig. 1.

To be specific, our approach pivots on the idea that while rich-resource inputs (like future frames) are challenging to obtain in application, they are accessible during the training phase. This availability allows for their utilization in guiding a LR single-image input model to enhance performance. Our methodology improves model performance with rich-resource guidance in two fundamental aspects. Firstly, we consider the features extracted from inputs with rich resources as a form of prior information. To achieve this, we utilize a collected generalized dataset with rich resources as a reference dataset. When estimating the depth for a LR single-image input, we initially search for similar pixels from the reference dataset. These pixels, which represent objects with similar geometric relationships, can offer valuable prior information for the model. With this prior information, the single-input model can perform similarly to richresource models. Secondly, we investigate the intrinsic consistency present in rich-resource model predictions. We observe that rich-resource models exhibit superior geometry consistency, particularly around object edges, compared to their LR single-image counterparts. Leveraging this consistency information en-

RprDepth 3



Fig. 1: Our main motivation. In self-supervised monocular depth estimation, models using rich-resource inputs generally achieve better performance. We aim to extract prior data from rich-resource inputs during offline training, using it to enhance models with single images.

hances the performance of the LR single-image model, especially in areas where depth estimation is traditionally challenging.

In addition, we propose a feature selection algorithm to reduce the computation burden of searching reference features during inference. This algorithm effectively reduces the search space for the appropriate prior features while maintaining the same performance. Experimental results demonstrate that our method can achieve similar performance to rich-resource models when only LR single-image inputs are available. This increases the feasibility of using the depth estimation method in real-world applications. Our contributions can be summarized in four folds:

- We propose a new approach for self-supervised monocular depth estimation that reduces the necessity for rich-resource, such as high-resolution, multiframe and future frame data, while still achieving superior performance compared to models that depend on such inputs.
- We propose incorporating a Prior Depth Fusion Module to effectively utilize the prior information obtained from rich-resource inputs.
- We propose the Rich-resource Guided Loss by considering the depth prediction from rich-resource inputs as a pseudo label. This approach harnesses the consistency embedded in the pseudo label to enhance the quality of the LR single-image model.
- We introduce an attention-guided feature selection algorithm to reduce the computation of searching for prior depth information during inference. With

this improvement, our model can achieve state-of-the-art performance while maintaining high processing speed with only LR single-image inputs.

2 Related Work

2.1 Supervised Monocular Depth Estimation

Supervised monocular depth estimation remains a core focus in computer vision, particularly for its applications in areas like autonomous vehicles and robotic navigation. This method relies on single images to infer depth maps, where each pixel value corresponds to the distance from the camera lens. In this domain, the supervised approach [1, 3, 12, 28, 28, 35, 36, 38, 41] necessitates ground truth depth data for training, presenting both opportunities and challenges.

The groundwork in this field was proposed by Eigen and colleagues [4], who innovatively utilized a deep learning model for depth prediction under supervised conditions. Their model's architecture featured a dual network setup, one for coarser depth perception and another for capturing fine-grained depth details. Following this pioneering work, several researchers have contributed to refining this approach. For instance, Li *et al.* [16] introduced the use of conditional random fields to enhance depth predictions, providing a new dimension to the estimation process.

Further explorations in geometry-based methods were conducted by Qi *et al.* [24], who proposed separate networks for estimating depth and surface normals from images. Ummenhofer *et al.* [29] contributed significantly with a network that predicts depth maps using structure from motion techniques. These advancements showcase a growing sophistication in the field. However, relying on extensive ground truth data, it is usually acquired through specialized equipment like LiDAR, limits the scalability and cost-effectiveness of these methods, and presenting an ongoing challenge for widespread application.

2.2 Self-supervised Monocular Depth Estimation

To mitigate the challenges associated with labeled data in monocular depth estimation, Garg *et al.* [6] pioneered a self-supervised learning methodology. This approach used stereo images during training, aiming to minimize the disparity between synthesized and real images, marking a significant shift from traditional supervised methods.

Building upon this, Zhou *et al.* [6] introduced a novel technique that estimated both the depth map and camera pose using single-camera video sequences. This method enabled the creation of artificial frames, facilitating the computation of disparities with real frames. However, this approach faced challenges such as occlusion and the presence of moving objects, which impacted the accuracy of depth estimation. Addressing these issues, Godard *et al.* [7] introduced a new minimum loss approach, exploiting the complementary nature of occlusions in adjacent frames. This allowed the model to selectively compute losses in visible areas, enhancing the accuracy of depth predictions. To address the moving object problem, they devised a strategy to ignore loss values from such objects, further refining the depth estimation process.

Subsequent research in this area has seen a variety of innovative approaches [8, 15, 21, 23, 25, 26, 42, 43]. Masoumian *et al.* [20] developed a multi-scale monocular depth estimation method using graph convolutional networks, offering a new perspective in this field. Guizilini *et al.* [9] proposed a 3D packing network, introducing a novel architecture in depth estimation. Watson *et al.* [33] incorporated cost volumes to build a multi-frame model, demonstrating significant improvements in depth accuracy. Furthermore, Zhou and colleagues [39] explored the integration of semantic information to enhance depth estimation, indicating the potential of combining different data types for improved results.

Despite these advancements, most of the best performance models in this area rely on rich-resource data as input, which limits their application in scenarios where capturing rich-resource data is difficult. This has motivated us to develop a depth estimator that utilizes only low-resolution single-image data but still produces highly accurate depth maps.

3 Method

In this section, we will provide information on the proposed Rich-resource Prior Depth estimator (RPrDepth). In Sec. 3.1, we will explain the pipeline of the proposed method and how it can be trained end-to-end. In Sec. 3.2, we will introduce the core module of our method, the Prior Depth Fusion Module. Then in Sec. 3.3, we will provide detailed information about the Rich-resource Guided Loss and how it guides the optimization of the model prediction. Finally, in Sec. 3.4, we will discuss the attention-guided feature selection algorithm for reducing computation in the feature searching during the inference phase.

3.1 Rich-resource Prior Depth Estimator

Rich-resource inputs, such as high-resolution images, multi-frame inputs and future frames, are valuable for the depth estimation task. They provide more information compared to single-frame low-resolution images, which we refer to as LR single-image inputs in this paper. However, in real-world scenarios, these rich-resource inputs are not always available, limiting the application of methods that rely on them. To address this issue, we propose a LR single-image depth estimator named Rich-resource Prior Depth estimator that bridges the performance gap between the two types of input data.

From a general perspective, LR single-image input cannot achieve the same performance as rich-resource inputs, as they lack the critical information encoded in the rich-resource inputs. For instance, when using multi-frame images as inputs, the model leverages the disparity between adjacent frames. However, LR single-image inputs do not possess this information and therefore cannot directly achieve similar performance. In this paper, we propose searching for the necessary information from the archived rich-resource inputs to bridge the



Fig. 2: Illustration of the Training Phase of Our Pipeline. Our pipeline comprises two branches: rich-resource and LR single-image. The former generates precise depth maps and features from rich-resource images, while the latter leverages these features to achieve comparable performance.

information gap between the two types of inputs. To be specific, we prepare a sub-dataset called *ref-dataset* which consists of rich-resource data with a wide range of variations. When we receive a LR single-image image, we search for similar feature pixels from the *ref-dataset*. These pixels come from similar objects with similar geometry relationships, but they contain rich-resource data. We can use this data to fill in the missing information in the LR single-image.

Fig. 2 provides an overview of the training pipeline for our Rich-resource Prior Depth estimator. The pipeline consists of two branches: the upper branch is proposed for rich-resource guidance, while the lower branch represents the LR single-image model. Our method is designed to be general-purpose, allowing for the use of a multi-frame model such as [32] or a high-resolution based model [7] for the rich-resource guidance. During the training phase, the rich-resource model remains fixed without gradient computation.

When training the model, we begin by selecting two distinct batches I_r, I_s from the *ref-dataset* and the LR single-image training dataset. Next, we calculate the image features using the rich-resource encoder (Encoder_r) and the LR single-image encoder (Encoder_s), respectively:

$$f_r = \text{Encoder}_{\mathbf{r}}(I_r); F_s = \text{Encoder}_{\mathbf{s}}(I_s).$$
(1)

After that, we use a convolution module to adjust the dimension of f_r to match that of F_s :

$$F_r = \operatorname{Conv}_{\mathbf{m}}(f_r). \tag{2}$$

To identify the most similar pixel in the reference dataset, we calculate the affinity between the target pixels and the reference pixels:

$$\mathcal{A} = \text{Softmax}(F_s \otimes F_r). \tag{3}$$



Fig. 3: Illustration of the Prior Depth Fusion Module.

Then, we generate the rich-resource depth map D_r by passing f_r into the richresource depth decoder:

$$D_r = \text{Decoder}_{\mathbf{r}}(f_r). \tag{4}$$

To efficiently extract and fuse the critical prior information encoded in the reffeatures, we propose the Prior Depth Fusion Module. This module takes \mathcal{A} , f_r , and D_r as input and produces a prior information-rich feature F_o as output. Finally, the values of F_o are passed to the LR single-image decoder to generate depth predictions D_o :

$$D_o = \text{Decoder}_{s}(F_o). \tag{5}$$

Finally, these predictions are used to construct the Rich-resource Guided Loss function. The entire pipeline is trained in an end-to-end manner using this loss function. Notably, the mentioned pipeline, which accepts both rich-resource and LR single-image inputs, is only used during the training phase. In the inference phase, the pipeline is adjusted to accept only LR single-image inputs, as explained in Sec. 3.4.

3.2 Prior Depth Fusion Module

In our Prior Depth Fusion Module, we have designed two types of fusion procedures to effectively extract and fuse features from the ref-dataset. These procedures are the pixel-wise fusion and the depth-hint fusion. The pixel-wise fusion is responsible for completing the missing prior information in LR single-image data using the corresponding rich-resource data as a reference. To efficiently identify the most similar pixel, we add an auxiliary loss to guide the search process. On the other hand, the depth-hint fusion aggregates the prior information from the entire ref-dataset in an attention manner, without any explicit guidance.

Fig. 3 shows an illustration of the Prior Depth Fusion Module. In this module, we first use a transformer module to extract and fuse the depth-hint prior information from the reference features. In this transformer, we consider the reference feature f_r as the key K and value V, and the target feature F_s as the query Q. Then we employ the multi-head attention to fuse the depth-hint information and produce the output feature F_d :

$$F_d = \mathrm{MHA}(\mathrm{Q}, \mathrm{K}, \mathrm{V}). \tag{6}$$

Next, we need to address the pixel-wise prior information. In the pipeline, we have calculated the affinity between the target and reference pixels. Using the affinity, f_r and D_r , we can construct a pixel-wise constructed reference feature map F_c and constructed reference depth D_c :

$$F_c = \mathcal{A} \times f_r; D_c = \mathcal{A} \times D_r.$$
(7)

Afterwards, we combine F_c and F_s and apply a convolution module to compress the feature map to its original number of channels. In addition to the reference features, we also take into account the output depth of the rich-resource model as valuable prior information. Consequently, we regard the prior depth map D_c as a reference and merge it with the feature map, similar to the reference feature, as shown in the figure. Furthermore, the constructed depth map is utilized to formulate an auxiliary loss. In this process, we minimize the discrepancy between the constructed depth map and the prediction of the high-resource model. This loss function can aid in guiding the optimization of the affinity matrix.

Notably, the Prior Depth Fusion Module involves calculating the attention matrix between the target batch and the reference batch, an operation with a space complexity of O(MN), where M and N are the pixel numbers of the target and reference batches, respectively. To enhance the representation of the reference data, we should use a relatively large size for the reference batch. However, this will result in a significant memory burden. To address this limitation and make the pipeline easier to train, we randomly sample features from the whole reference dataset offline (1% pixels from 2000 images). This is based on the observation that adjacent pixels have similar geometric information. Therefore, when we randomly sample from a large batch, the selected pixels can provide more contextual conditions than selecting all pixels from a smaller batch.

3.3 Rich-resource Guided Loss

To enhance the performance of the proposed pipeline and the Prior Depth Fusion Module, we recommend incorporating the Rich-resource Guided Loss, as shown in Fig. 4 (a). This loss function effectively utilizes guidance from both rich-resource inputs and the rich-resource model predictions to optimize the LR single-image model. The proposed loss function consists of two parts: the viewpoint reconstruction loss guided by rich-resource inputs and the consistency loss guided by the predictions of the rich-resource model.

Most of the self-supervised monocular depth estimation methods use the viewpoint reconstruction loss to guide the model optimization. Following the previous method [7], we also use the viewpoint reconstruction as the main loss function. As we have rich-resource inputs available during the training phase, we choose to reconstruct the new viewpoint images from these inputs. These inputs contain more detailed information and can provide more accurate guidance for

RprDepth 9



Fig. 4: Illustration of the Loss and Inference Pipeline. (a) Illustration of the Rich-resource guided loss. (b) Illustration of Attention Guided Feature Selection. (c) The Inference Pipeline of RPrDepth.

the model. To bridge the resolution gap between our prediction and the richresource inputs, we upsample the prediction with cubic interpolation algorithm to match the size of the target image, and the final reconstruction loss is defined as:

$$\mathcal{L}_{\rm vp} = l_{vp} \left(\text{Resize}(D_o), I_r \right) \tag{8}$$

In addition to the reconstruction loss, we also leverage the pseudo labels generated by the rich-resource model to optimize the LR single-image model. rich-resource models typically produce depth maps with greater detail accuracy, particularly in the edge areas, compared to ordinary LR single-image depth estimators. Therefore, it is meaningful to utilize the advantages of rich-resource predictions to instruct the LR single-image model. However, directly using the pseudo label as the target and minimizing the difference between the predictions and the pseudo labels is not a desirable approach. Since the models are trained in a self-supervised manner, their predictions represent relative disparity rather than accurate depth values. As a result, different models may have variations in scale in their predictions. Hence, we choose to minimize the gradients along the x and y axes between the prediction and the rich-resource predictions.

Specifically, we start by calculating the gradient on the x and y-axis of the output depth map D_o and the pseudo label generated offline D_p . It's important to note that despite both D_p and D_r being depth maps generated by the rich-resource model, they're derived from different batches. Specifically, D_r comes from the reference batch, while D_p is from the training batch. Next, we normalize

the two gradient maps and add up the x and y-gradients.

$$G_{x,y}(D_o) = \operatorname{Norm}(\nabla_x D_o) + \operatorname{Norm}(\nabla_y D_o)$$

$$\tilde{G}_{x,y}(D_p) = \operatorname{Norm}(\nabla_x D_p) + \operatorname{Norm}(\nabla_y D_p).$$
(9)

We use an L1 loss to minimize the gradient difference:

$$\mathcal{L}_{c} = \|\hat{G}_{x,y}(D_{o}) - \hat{G}_{x,y}(D_{p})\|_{1}.$$
(10)

Additionally, as mentioned in the previous section, we use an auxiliary loss L_{aux} to guide the optimization of the affinity matrix. To achieve this, we up-sample the constructed depth-map D_c to the same size as the pseudo label D_p . We then minimize the difference between them directly. Since D_c is constructed directly from the pixels of the high-resource model prediction, it must have the same scale factor as D_p . Therefore, we can simply minimize their difference rather than the gradient:

$$\mathcal{L}_{\text{aux}} = \|D_p - \text{Resize}(D_c)\|_1.$$
(11)

The final loss is determined by the combination of the reconstruction loss, the consistency loss and the auxiliary loss:

$$\mathcal{L} = \alpha \mathcal{L}_{\rm vp} + \beta \mathcal{L}_{\rm c} + \mathcal{L}_{\rm aux}, \tag{12}$$

where α, β are the balance ratios.

3.4 Attention Guided Feature Selection

As mentioned in the previous sections, the number of the reference dataset is crucial. The reference dataset should have sufficient variety to encompass all possible conditions that may be found in the target image. However, if the size of this reference dataset is too large, it may result in a significant computational burden when searching for the reference pixels. To overcome this limitation, we propose a new solution that involves using a subset of the reference dataset instead of the entire dataset during the inference phase. This subset is selected to be the most representative of the reference dataset. To achieve this, we introduce the attention-guided feature selection algorithm, as shown in Fig. 4 (b). The proposed algorithm selects the features from the reference dataset in an attention-based manner.

In the depth-hint fusion procedure, the reference features are used with multihead attention, while in the pixel-wise fusion procedure, the features are incorporated with a learnable affinity matrix. By leveraging these two weight matrices, we can determine which pixels are more important for the target image. We then aggregate the weight matrices across the entire validation dataset and calculate the average weight matrix for each pixel in the reference dataset:

$$W_{\text{avg}} = \frac{1}{N} \sum_{i=1}^{N} (\mathcal{A}_i + \mathcal{A}_{\text{MHA},i})$$
(13)

Finally, we sort the pixels in the matrix and select the ones with the highest weight to represent the entire reference dataset.

We store the pixels with the highest weight value. These pixels serve as the depth-prior data, which remains unchanged during the inference phase. Once the depth-prior data is generated, we replace it with the original rich-resource model in our pipeline and fine-tune the LR single-image model for a few epochs. Surprisingly, we find that the performance of the LR single-image model does not decrease due to the decrease in computation, but actually outperforms the original model. We attribute this to the fact that the selected pixels are more concise and meaningful, contributing to the improved performance. By eliminating other irrelevant pixels that may cause interference, the model can better learn to utilize this prior information.

The final inference pipeline is shown in Fig. 4(c). In comparison to the original ref-dataset based pipeline, the compressed prior data is only less than 1% of the original size (from 2,560,000 to 25,000 pixels), significantly reducing the computational load.

4 Experiment

For all the training and evaluation processes, we utilize one work station with a single V100 GPU. To demonstrate the enhancements, we integrate these advancements alongside a recent, highly efficient baseline known as DIFFNet [39], inspired by the HR-Net networks [19, 27]. For the high-resource guidance, we have opted for ManyDepth [32] as our choice. ManyDepth is a well-known baseline model that utilizes multi-frame images as input. During the training of our model, we specifically chose the HR version of ManyDepth to provide the feature prior and loss guidance. This model accepts multi-frame, high-resolution images along with future frames as input.

4.1 Comparison on KITTI

The KITTI dataset stands out as a widely utilized benchmark in the field of computer vision. It's also highly regarded as a benchmark in the area of selfsupervised monocular depth estimation. Our approach utilizes the data partitioning strategy mentioned in [4] as a foundation for our models, and we follow the preprocessing steps outlined in [40] to eliminate static frames. During the training phase, we randomly select 2,000 triplets from the training dataset as the reference dataset, and use the remaining 37, 810 triplets as the training data. When running the feature selection procedure, we use a separated validation set as the target, ensuring that it has no overlap with the test set.

SOTA comparison Table 1 presents our RPrDepth's assessment on the Eigen split [4], categorizing results by low and high resolutions. We utilize seven metrics for comparison, with *AbsRel*, *SqRel*, *RMSE*, *RMSElog* as error metrics where lower scores indicate better performance. Conversely, δ measures the deviation from actual depth values, with $\delta < 1.25$, $\delta < 1.25^2$, $\delta < 1.25^3$ being accuracy

Table 1: The SOTA comparison on KITTI Eigen Split [4]. We evaluate our methods against established models on this benchmark, using three self-supervision techniques: "M" for monocular videos, "S" for stereo images, and "MS" for both. The best and second-best results are marked in **bold** and <u>underline</u>, respectively.

Method	TestFrames	Resolution	Trian	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [7]	1	640×192	M	0.115	0.903	4.863	0.193	0.877	0.959	0.981
PackNet-SfM [9]	1	640×192	M	0.111	0.785	4.601	0.189	0.878	0.960	0.982
HR-Depth [19]	1	640×192	M	0.109	0.792	4.632	0.185	0.884	0.962	0.983
DIFFNet [39]	1	640×192	M	0.102	0.764	4.483	0.180	0.896	0.965	0.983
BRNet [34]	1	640×192	M	0.105	0.698	4.462	0.179	0.890	0.965	0.984
MonoFormer [2]	1	640×192	M	0.108	0.806	4.594	0.184	0.884	0.963	0.983
Lite-Mono [37]	1	640×192	M	0.107	0.765	4.561	0.183	0.886	0.963	0.983
Wang et al. [30]	2(-1, 0)	640×192	M	0.106	0.799	4.662	0.187	0.889	0.961	0.982
ManyDepth [32]	2(-1, 0)	640×192	M	<u>0.098</u>	0.770	4.459	0.176	0.900	0.965	0.983
RPrDepth (ours)	1	640×192	M	0.097	0.658	4.279	0.169	0.900	0.967	0.985
Monodepth2 [7]	1	640×192	S	0.109	0.873	4.960	0.209	0.864	0.948	0.975
BRNet [34]	1	640×192	S	<u>0.103</u>	0.792	4.716	0.197	<u>0.876</u>	0.954	0.978
RPrDepth (ours)	1	640×192	S	0.098	0.716	4.538	0.185	0.885	0.960	0.980
HR-Depth [19]	1	640×192	MS	0.107	0.785	4.612	0.185	0.887	0.962	0.982
DIFFNet [39]	1	640×192	MS	0.101	0.749	4.445	0.179	<u>0.898</u>	0.965	<u>0.983</u>
BRNet [34])	1	640×192	MS	<u>0.099</u>	0.685	4.453	0.183	0.885	0.962	<u>0.983</u>
RPrDepth (ours)	1	640×192	MS	0.095	0.638	4.232	0.169	0.902	0.970	0.985
PackNet-SfM [9]	1	1280×384	M	0.107	0.802	4.538	0.186	0.889	0.962	0.981
HR-Depth [19]	1	1024×320	M	0.106	0.755	4.472	0.181	0.892	0.966	0.984
DIFFNet [39]	1	1024×320	M	0.097	0.722	4.435	0.174	0.907	0.967	0.984
BRNet [34]	1	1024×320	M	0.103	0.684	4.385	0.175	0.889	0.965	0.985
Lite-Mono [37]	1	1024×320	M	0.102	0.746	4.444	0.179	0.896	0.965	0.983
Wang et al. [30]	2(-1, 0)	1024×320	M	0.106	0.773	4.491	0.185	0.890	0.962	0.982
ManyDepth-HR [32]	2(-1, 0)	1024×320	M	<u>0.093</u>	0.715	4.245	0.172	<u>0.909</u>	0.966	0.983
RPrDepth (ours)	1	1024×320	Μ	0.091	0.612	4.098	0.162	0.910	0.971	0.986
Monodepth2 [7]	1	1024×320	S	0.107	0.849	4.764	0.201	0.874	0.953	0.977
BRNet [34]	1	1024×320	S	<u>0.097</u>	0.729	4.510	0.191	0.886	0.958	0.979
RPrDepth (ours)	1	1024×320	S	0.091	0.689	4.412	0.185	0.892	0.959	0.979
HR-Depth [19]	1	1024×320	MS	0.101	0.716	4.395	0.179	0.899	0.966	0.983
BRNet [34]	1	1024×320	MS	0.097	0.677	4.378	0.179	0.888	0.965	0.984
DIFFNet [39]	1	1024×320	MS	0.094	0.678	4.250	0.172	<u>0.911</u>	0.968	0.984
BPrDepth (ours)	1	1024×320	MS	0.089	0.613	4 1 2 0	0.159	0.913	0.970	0.985

metrics where higher scores are favorable. Our RPrDepth tops all categories in terms of supervision types and resolutions.

As shown in this table, our model with LR single-image input outperforms our baseline model DIFFNet, and even outperforms the guiding model ManyDepth-HR, which is based on multi-frame high-resolution inputs. Notably, the performance of ManyDepth in this table is without future frames, because future frames are not available during inference.

Qualitative Results Fig. 5 shows a comparison between our model, DIFFNet, and the guiding model ManyDepth. In comparison to models with rich-resource input, our model performs better on moving objects, as demonstrated in Fig. 5 (a). This is because multi-frame based methods are not well-suited for moving objects [32]. However, our model can identify relevant information for moving objects and correct the issue. Additionally, compared to other single image models, our model can address incorrect depth predictions caused by texture, as seen in Fig. 5 (c) with the arrow on the road. Ordinary LR single-image models struggle to distinguish texture, but our model leverages prior information from rich references to solve this problem.



Fig. 5: Qualitative results on the KITTI Eigen split test set. Our RPrDepth can correct the errors of both LR single-image models and rich-resource based models.

Architecture	Abs Rel \downarrow	Sq Rel \downarrow	RMSE↓	$log_{10}\downarrow$
Monodepth2	0.322	3.589	7.414	0.163
BRNet	0.302	3.133	7.068	0.156
RPrDepth	0.288	2.868	6.532	0.145

Table 2: Make3D results with monocular training and 640×192 inputs.

4.2 Comparison on Make3D and Cityscapes

Make3D dataset is composed of both single-camera RGB images and their related depth maps. It lacks stereo images and monocular sequences, rendering it unsuitable for training self-supervised monocular depth estimation models. However, it is commonly used as a test set to assess the performance of networks on a varied dataset. In our study, we evaluated our models against other notable research in this area. The results, as detailed in Table 2, reveal that our models surpass all competing methods, indicating their robustness in adapting to novel environments. Utilizing monocular training with a resolution of 640×192 , our approach records 0.288 in *AbsRel* and 6.532 in *RMSE*, markedly surpassing other leading models in performance.

Citiscapes dataset stands as a key resource in the field of semantic segmentation, particularly for autonomous driving applications. It encompasses a collection of stereo video sequences, which are instrumental for training self-supervised depth estimation models. Adhering to the approach outlined in [33], we conducted training and evaluation of our RPrDepth model using the Cityscape dataset. The outcomes, as presented in Table 3, demonstrate that RPrDepth remarkably exceeds the performance of numerous advanced models.

4.3 Ablation Study

We performed several ablation studies on the KITTI dataset. We used the Eigen split [4] to validate the effectiveness of the proposed modules: Prior Depth Fusion (PDF) module, Attention Guided Feature Selection (AGFS), and Rich-resource

Architecture	Frames	Abs Rel \downarrow	RMSE↓	$\delta < 1.25 \uparrow$
Monodepth2 [7]	1	0.129	6.876	0.849
Li et al. [17]	1	0.119	6.980	0.846
ManyDepth [32]	2(-1,0)	0.114	6.223	<u>0.875</u>
RPrDepth	1	0.111	6.243	0.890

Table 3: Cityscape results follow the settings of [33].

Table 4: Ablation study of the proposed RPrDepth.

Components	Abs Rel \downarrow	$RMSE \downarrow$	$\delta < 1.25 \uparrow$
Baseline	0.102	4.483	0.896
+ PDF	0.098	4.284	0.898
+ AGFS	0.098	4.240	0.898
+ RGL	0.100	4.321	0.897
+ Full	0.097	4.279	0.900

Guided Loss (RGL). Specifically, +PDF indicates the baseline model with richresource feature prior, +AGFS module indicates that we replaced the reference dataset with the selected features, and +RGL indicates that the model was trained with the proposed new loss function. Lastly, +Full indicates the model with all the components.

As shown in Table 4, integrating prior information from rich resource data significantly improves the model across all metrics. After applying the feature selection algorithm, we further enhance the performance, particularly in terms of the RMSE metric. Additionally, the computational burden of the search process is significantly reduced. Overall, the feature selection algorithm reduces the number of features to just 1% of the entire reference dataset. The proposed RGL also clearly improves performance. Finally, the combination of all components achieves the best performance.

5 Conclusion

In the field of self-supervised depth estimation, many top-performing models use rich-resource images as input, such as multi-frame images and high-resolution images. However, these rich-resource inputs are not always available in real-world applications. Therefore, in this paper, we propose a new depth estimation model that leverages the prior information encoded in rich-resource images during the training and uses only a single image to generate the depth map during the inference phase. Specifically, we propose three key modules. The first module is the Prior Depth Fusion, which efficiently combines the prior features. The second module is the Rich-resource Guided Loss, which guides the optimization of LR single-image models. Lastly, we introduce the Attention Guided Feature Selection algorithm to enhance the searching efficiency from the reference images. We aim for our method to provide a new perspective on improving the practicality of high-performance depth estimation.

References

- 1. Ashutosh Agarwal and Chetan Arora. Depthformer: Multiscale vision transformer for monocular depth estimation with local global information fusion. *arXiv preprint* arXiv:2207.04535, 2022. 4
- Jinwoo Bae, Sungho Moon, and Sunghoon Im. Monoformer: Towards generalization of self-supervised monocular depth estimation with transformers. arXiv preprint arXiv:2205.11083, pages 1, 2, 4, 5, 6, 7, 2022.
- Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2183–2191, 2019.
- David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 4, 11, 12, 13
- Ziyue Feng, Liang Yang, Longlong Jing, Haiyan Wang, YingLi Tian, and Bing Li. Disentangling object motion and occlusion for unsupervised multi-frame monocular depth. In *European Conference on Computer Vision*, pages 228–244. Springer, 2022. 2
- 6. Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 4
- Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. 2, 4, 6, 8, 12, 14
- Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *ICCV*, pages 8977–8986, 2019. 5
- Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon.
 3d packing for self-supervised monocular depth estimation. In CVPR, 2020. 5, 12
- Vitor Guizilini, Rareş Ambruş, Dian Chen, Sergey Zakharov, and Adrien Gaidon. Multi-frame self-supervised depth with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 160– 170, 2022. 2
- Wencheng Han, Junbo Yin, and Jianbing Shen. Self-supervised monocular depth estimation by direction-aware cumulative convolution network. In *Proceedings of* the *IEEE/CVF International Conference on Computer Vision*, pages 8613–8623, 2023. 2
- 12. Tak-Wai Hui. Rm-depth: Unsupervised learning of recurrent monocular depth in dynamic scenes. In CVPR, 2022. 4
- Varun Ravi Kumar, Senthil Yogamani, Markus Bach, Christian Witt, Stefan Milz, and Patrick M\u00e4der. Unrectdepthnet: Self-supervised monocular depth estimation using a generic framework for handling common camera distortion models. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 8177–8183. IEEE, 2020. 2
- 14. Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020. 1
- Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326, 2019. 5
- 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, 2015. 4

- 16 W. Han et al.
- Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. In *Conference on Robot Learning*, pages 1908–1917. PMLR, 2021. 14
- Kun Liu, Changhe Zhou, Shengbin Wei, Shaoqing Wang, Xin Fan, and Jianyong Ma. Optimized stereo matching in binocular three-dimensional measurement system using structured light. *Applied optics*, 53(26):6083–6090, 2014.
- Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: high resolution self-supervised monocular depth estimation. CoRR abs/2012.07356, 2020. 2, 11, 12
- Armin Masoumian, David GF Marei, Saddam Abdulwahab, Julian Cristiano, Domenec Puig, and Hatem A Rashwan. Absolute distance prediction based on deep learning object detection and monocular depth estimation models. In CCIA, pages 325–334, 2021. 5
- Sudeep Pillai, Rareş Ambruş, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *ICRA*, 2019. 5
- 22. Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *CVPR*, 2020. 2
- Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In 3DV, 2018. 5
- Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In CVPR, 2018. 4
- René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zeroshot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 5
- Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In CVPR, 2019.
 5
- Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. arXiv preprint arXiv:1904.04514, 2019.
 11
- 28. Kunal Swami, Amrit Muduli, Uttam Gurram, and Pankaj Bajpai. Do what you can, with what you have: Scale-aware and high quality monocular depth estimation without real world labels. In *CVPR*, 2022. 4
- Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In CVPR, 2017. 4
- Jianrong Wang, Ge Zhang, Zhenyu Wu, XueWei Li, and Li Liu. Self-supervised joint learning framework of depth estimation via implicit cues. arXiv preprint arXiv:2006.09876, 2020. 12
- Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *ICCV*, pages 2162–2171, 2019.
 2
- 32. Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1164–1174, 2021. 2, 6, 11, 12, 14

- Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In CVPR, 2021. 5, 13, 14
- Han Wencheng, Yin Junbo, Jin Xiaogang, Dai Xiangdong, and Shen Jianbing. Brnet: Exploring comprehensive features for monocular depth estimation. In ECCV, 2022. 12
- 35. Chi Xu, Baoru Huang, and Daniel S Elson. Self-supervised monocular depth estimation with 3-d displacement module for laparoscopic images. *IEEE Transactions* on Medical Robotics and Bionics, 4(2):331–334, 2022. 4
- 36. Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18537–18546, 2023. 4
- 37. Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18537–18546, 2023. 12
- 38. Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. In 2022 International Conference on 3D Vision (3DV), pages 668–678. IEEE, 2022. 4
- Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. In *British Machine Vision Conference* (*BMVC*), 2021. 5, 11, 12
- 40. Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 11
- Zhengming Zhou and Qiulei Dong. Self-distilled feature aggregation for selfsupervised monocular depth estimation. In ECCV, pages 709–726. Springer, 2022.
 4
- Zhongkai Zhou, Xinnan Fan, Pengfei Shi, and Yuanxue Xin. R-msfm: Recurrent multi-scale feature modulation for monocular depth estimating. In *ICCV*, 2021. 5
- 43. Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, 2018. 5