Machine Vision and Applications (2024) 35:115
https://doi.org/10.1007/s00138-024-01586-4

RESEARCH

Dyna-MSDepth: multi-scale self-supervised monocular depth estimation network for visual SLAM in dynamic scenes

Jianjun Yao[1] · Yingzhao Li[1] · Jiajia Li[1]

Received: 21 May 2023 / Revised: 24 March 2024 / Accepted: 11 July 2024 / Published online: 19 August 2024

Abstract
Monocular Simultaneous Localization And Mapping (SLAM) suffers from scale drift, leading to tracking failure due to scale ambiguity. Deep learning has significantly advanced self-supervised monocular depth estimation, enabling scale drift reduction. Nonetheless, current self-supervised learning approaches fail to provide scale-consistent depth maps, estimate depth in dynamic environments, or perceive multi-scale information. In response to these limitations, this paper proposes Dyna-MSDepth, a novel method for estimating multi-scale, stable, and reliable depth maps in dynamic environments. Dyna-MSDepth incorporates multi-scale high-order spatial semantic interaction into self-supervised training. This integration enhances the model's capacity to discern intricate texture nuances and distant depth cues. Dyna-MSDepth is evaluated on challenging dynamic datasets, including KITTI, TUM, BONN, and DDAD, employing rigorous qualitative evaluations and quantitative experiments. Furthermore, the accuracy of the depth maps estimated by Dyna-MSDepth is assessed in monocular SLAM. Extensive experiments confirm the superior multi-scale depth estimation capabilities of Dyna-MSDepth, highlighting its significant value in dynamic environments. Code is available at https://github.com/Pepper-FlavoredChewingGum/Dyna-MSDepth.

Keywords Monocular depth estimation · Dynamic scenes · Visual SLAM · Self-supervised learning · Scale drift

1 Introduction

Visual SLAM is vital for autonomous robots, with applications ranging from autonomous driving [1] to 3D reconstruction [2], navigation [3], and high-definition mapping [4]. Monocular SLAM, enabled by low-cost cameras, is crucial in this domain [5]. However, scale ambiguity often leads to significant scale drift during long-term operation, particularly in dynamic scenes [6].

Addressing scale drift in monocular SLAM usually involves providing depth information for image pixels [7]. Traditionally, this required additional sensors like stereo cameras or LiDAR [8], which increases costs and complexity. Recent advances in monocular depth estimation networks offer a promising alternative. While supervised methods require expensive LiDAR for training, self-supervised approaches, like Hr-depth [9] and Monodepth2 [10], learn from geometric projection matching without depth ground truth.

Nevertheless, most monocular depth estimation networks failed to produce consistent depth maps between frames, leading to tracking failures in SLAM [11]. In dynamic scenes, the depth estimation of dynamic objects was highly inaccurate and their edges appeared blurred, primarily attributed to the violation of the geometric consistency assumption [12]. Moreover, monocular estimation methods lacked accurate scale estimation and struggled to perceive distance between objects and the camera, leading to increased inaccuracy in depth estimation for dynamic objects, particularly in scenes with low discriminability, such as highways [7].

This paper proposes Dyna-MSDepth, a novel self-supervised monocular depth estimation network, aimed at addressing the aforementioned challenges. As shown in Fig. 1, Dyna-MSDepth produces a stable and reliable depth map with

✉ Jianjun Yao
travisyao@hrbeu.edu.cn

Yingzhao Li
yzli@hrbeu.edu.cn

Jiajia Li
hahajaja@hrbeu.edu.cn

[1] College of Mechanical and Electrical Engineering, Harbin Engineering University, Nantong Street, Harbin 150001, Heilongjiang, China

sharp edge segmentation and scale consistency in dynamic scenes, enabling direct utilization for monocular SLAM scale recovery. Dyna-MSDepth employs scale consistency loss to establish connections between depth values in consecutive frames. Additionally, a pre-trained supervised model provides depth priors, enabling depth value prediction on dynamic objects and segmentation of dynamic object edges. To perceive objects at varying distances, multi-scale input is employed in the monocular sequences, while high-order spatial interaction enhances feature fusion. Dyna-MSDepth is extensively evaluated on four challenging dynamic datasets, including KITTI, TUM, DDAD, and BONN. The results demonstrate that Dyna-MSDepth outperforms existing State-Of-The-Art (SOTA) approaches both qualitatively and quantitatively. Moreover, the dense depth maps estimated by Dyna-MSDepth are directly utilized for scale recovery in ORB-SLAM3, resulting in significantly reduced scale drift in the KITTI dataset.

In summary, the contributions of this paper are:

1. Dyna-MSDepth is proposed as a solution for multi-scale monocular depth estimation in dynamic scenes, enabling direct utilization of the estimated depth maps for scale recovery in monocular SLAM to mitigate scale drift;
2. The performance of Dyna-MSDepth is assessed through qualitative and quantitative evaluations on four challenging dynamic datasets, i.e., KITTI, TUM, DDAD, and BONN;
3. The effectiveness of Dyna-MSDepth's depth map is demonstrated through its application in monocular SLAM, with evaluations conducted on the KITTI datasets.

2 Related work

This chapter presents an overview of the relevant researches. Firstly, the scale drift problem in monocular SLAM is addressed (Sect. 2.1). Secondly, the current research status of monocular depth estimation networks is analyzed (Sect. 2.2). Finally, the significance of multi-scale approaches in self-supervised monocular depth estimation is discussed (Sect. 2.3).

2.1 Scale drift in monocular SLAM

Visual SLAM encompasses monocular [7], stereo [14], and RGBD [15] techniques, each employing distinct sensor setups. Monocular SLAM is preferred due to its cost-effectiveness and simplicity [5]. Nonetheless, it suffers from scale ambiguity, leading to significant scale drift over time. Consequently, this limitation severely impacts subsequent tasks such as localization, mapping, 3D reconstruction, and navigation [16]. To address the single-purpose scale drift, [8] and [17] incorporated ridar and IMU sensors. [18] assumed a known camera height from the ground. [19] and [7] employed costly global bundle adjustment after calculating the Sim(3) transformation. Several learning-based visual odometry methods aimed to directly recover the absolute scale in the scene [20–22]. However, their accuracy lagged behind traditional multi-view geometry-based SLAM algorithms due to the absence of bundle adjustment, feature search, and loop closing [23].

2.2 Self-supervised monocular depth estimation

Monocular depth estimation methods in deep learning are categorized into three approaches. The first category involves supervised training using ground truth obtained from LiDAR and RGB-D cameras. [24] leveraged high-order three-dimensional geometric constraints and depth truth values to enhance depth prediction accuracy. [25] improved supervised training performance through the incorporation of attention mechanisms. These approaches necessitate costly depth truth data. The second category utilizes calibrated binocular cameras to project depth maps from the left to the right camera and subsequently calculates photometric loss. [26] addressed the uncertainties associated with stereo depth estimation. While this method can provide metric depth, it requires precise camera calibration. The third category explores unsupervised or weakly supervised methods to learn depth prediction. [10] was the earliest and most typical self-supervised monocular depth estimation method. [27] enhanced the performance of self-supervised method through semantic guidance. [28–30] investigated the impact of physical world attacks on self-supervised monocular depth estimation and proposed several effective countermeasures. The use of self-supervised methods, which do not rely on depth ground truth, is prevalent. To address the challenges of scale inconsistency, dynamic object estimation, and scale ambiguity in self-supervised monocular depth estimation, [11] had proposed scale consistency loss to ensure continuity of inter-frame depth values. Additionally, [12] had introduced pseudo depth approach to tackle the depth estimation problem of dynamic objects. These advancements contribute to the growing popularity of self-supervised monocular depth estimation.

2.3 Visual SLAM combined with depth estimation

Acquiring camera pose and maintaining global consistency mapping in dynamic environments poses significant challenges. Numerous methodologies have been developed to address this issue. [31] employed SegNet for semantic segmentation, effectively eliminated feature points within dynamic regions. [32] combined MaskRCNN with ORB-SLAM2 [19] to detect moving vehicles, thereby mitigating the impact of dynamic obstacles on static background pose estimations. [33] leveraged semantic information to aid

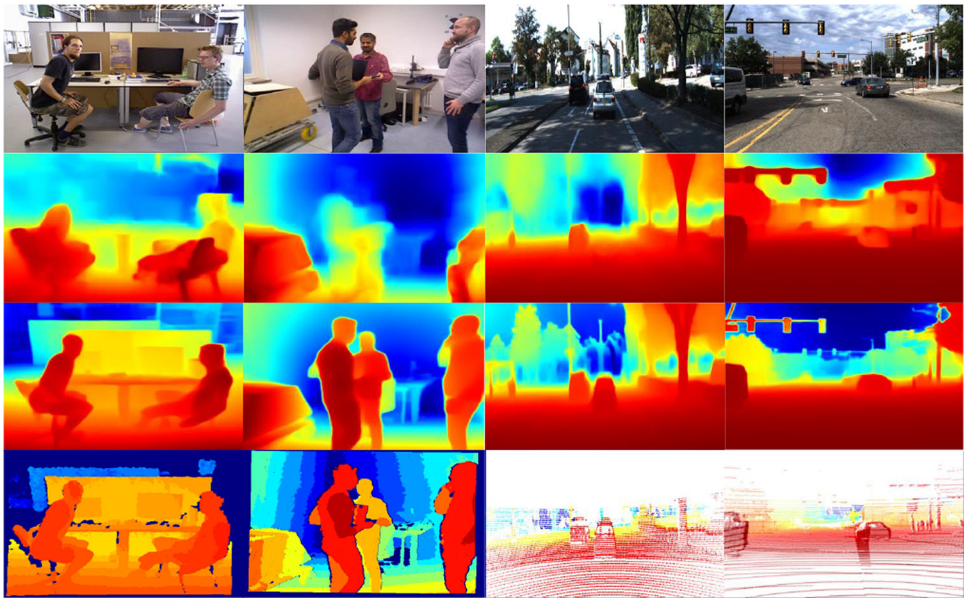

Fig. 1 Dyna-MSDepth is evaluated against SOTA methods on 4 challenging dynamic datasets. Top row: the original images from the TUM, BONN, KITTI, and DDAD datasets, respectively. Second row: the monocular depth estimation results of other SOTA methods. The left two columns are the results of [13], and the right two columns are the results of [11]. Third row: the monocular depth estimation results of Dyna-MSDepth. Bottom row: the ground truth depth maps provided by the datasets

in epipolar geometry calculation within dynamic environments, overcoming noise interference in motion information between adjacent frames. [34] integrated object detection and depth information for dynamic feature recognition, achieving performance levels comparable to semantic segmentation. Moreover, [34] employed IMU for motion prediction in feature tracking and motion consistency checking. [35] utilized the interdependence between camera motion and optical flow, optimizing them jointly within a unified learning framework in dynamic environments. [36] optimized static scenes, dynamic object structures, and camera pose simultaneously, facilitating the decoupling and estimation of three-dimensional bounding boxes for dynamic objects within fixed time windows. While existing methodologies effectively handle dynamic feature points and optical flow, enhancing positioning accuracy and robustness in dynamic environments, monocular SLAM invariably suffers from significant scale drift regardless of the approach employed.

2.4 Multi-scale models

Image pyramids are extensively employed in computer vision tasks to enable models to handle multi-resolution and multi-scale information [7, 19, 37]. Image pyramids simulate various object distances, which is crucial for handling the scale blurriness in monocular vision. It aids monocular depth estimation and SLAM in perceiving depth values for objects of different sizes. [38] employed multi-scale input and attention mechanism to enhance spatial perception in monocular depth estimation. [39–41] captured fine-grained images with multi-scale input. [42] mitigated the impact of object size differences on the Convolutional Neural Network (CNN) model through multi-scale input. While the CNN model establishes multi-scale through downsampling in its backbone, it pertains to the feature layer post complex operations rather than the original RGB image [43, 44]. By incorporating multi-scale input, Dyna-MSDepth effectively emulates objects at different distances and scales, thereby improving the accuracy of dynamic object depth estimation.

3 Method

This section presents the principles of Dyna-MSDepth. Firstly, Sect. 3.1 introduces the principle of self-supervised monocular depth estimation. Section 3.2 further analyzes the theory of depth ranking, which serves as the foundation for Dyna-MSDepth in estimating depth in dynamic scenes. Section 3.3 proposes a multi-scale depth estimation module that incorporates high-order spatial interaction, leading to improved performance compared to the baseline. Building upon these foundational theories, Sect. 3.4 introduces the specific architecture of Dyna-MSDepth for self-supervised multi-scale monocular depth estimation in dynamic scenes.

3.1 Self-supervised monocular depth estimation

The self-supervised monocular depth estimation approach consists of two components: DepthNet for depth estimation and PoseNet for 6D pose estimation [10]. During the training process, given a pair of monocular images (I_x, I_y), DepthNet estimates the dense depth map (D_x, D_y), while PoseNet estimates the relative pose P_{xy} between the image pair. Subsequently, D_y is employed to generate the simulated image I'_x of the previous frame using P_{xy} projection. By calculating the pixel difference between I_x and I'_x, the self-supervised monocular depth estimation method is trained.

Several monocular depth estimation networks lack the ability to produce a consistent and densely connected depth map, resulting in discontinuous depth values between frames [45, 46]. To address this issue, Dyna-MSDepth incorporates a consistency loss L_G from [11] to enforce the continuity of depth values across frames, thereby enhancing the stability of downstream tasks like SLAM. For the point p in D_y that is successfully projected to D_x, the geometric inconsistency of p between the synthetic depth map D_x^y and the target depth map D'_x can be calculated:

$$D_{\text{diff}} = \frac{\left\| D'_x(p) - D_x^y(p) \right\|_2}{D'_x(p) + D_x^y(p)}, \tag{1}$$

while D_x^y represents the depth map obtained by projecting D_y using P_{xy}, and D'_x denotes the depth map produced by interpolation aligned with D_x^y. Then, the geometric consistency loss L_G is calculated as:

$$L_G = \frac{1}{|U|} \sum_{p \in P} D_{\text{diff}}(p) = \frac{\left\| D'_x(p) - D_x^y(p) \right\|_2}{D'_x(p) + D_x^y(p)}, \tag{2}$$

where U represents the effective projection points. The L_G penalty was applied during the training process to ensure depth consistency within each batch, resulting in continuous depth maps for the entire image sequence.

Equation 2 normalizes depth map inconsistencies through summation. When the dynamic objects appear in training frames, loss L_G rapidly increases due to multi-view consistency assumption violation. Hence, Dyna-MSDepth cites a weight parameter [13]:

$$M_s = 1 - D_{\text{diff}} \tag{3}$$

The parameter M_s, ranging from 0 to 1, denotes the gradient proportion from the loss function L_G in training. A smaller A implies higher likelihood of dynamic objects in the region, necessitating decreased contribution to the loss function L_G.

Additionally, Dyna-MSDepth employs a photometric loss L_P from [11], which is weighted, to constrain I_x and I'_x:

$$\begin{aligned} L_P = \frac{1}{|U|} \sum_{p \in U} (\lambda \left\| I_x(p) - I'_x(p) \right\| \\ +(1-\lambda) \frac{1 - \text{SSIM}_{xx'}(p)}{2}), \end{aligned} \tag{4}$$

$$L_P^M = \frac{1}{|U|} \sum_{p \in U} (M_s(p) L_P(p)), \tag{5}$$

Finally, Dyna-MSDepth incorporates an edge-aware smoothing loss L_S from [13] to effectively regularize the depth map:

$$L_S = \sum_p \left(e^{-\nabla I_x(p)} \nabla D_x(p)\right)^2, \tag{6}$$

where ∇ denotes the first derivative with respect to the spatial dimension.

In cases where the dynamic object occupies a minority of image pixels, the loss function can include geometric consistency loss, photometric loss, and edge smoothing loss:

$$L_{\text{self}} = \alpha L_G + \beta L_P^M + \gamma L_S \tag{7}$$

The weights of three losses are denoted as α, β, and γ, respectively.

3.2 Dynamic region refinement

Supervised monocular depth estimation is not directly applicable for dynamic scenes, but it can assist unsupervised monocular depth estimation [27]. The supervised network, trained on large-scale datasets, offers advantages in depth ordinal, depth value smoothness, and sharp object edges. It particularly aids in training unsupervised monocular depth estimation networks in dynamic scenes, effectively capturing near-far point relationships of dynamic objects [12]. To address dynamic scenes and ensure fair comparisons, this study adopted the approach of [12] for handling dynamic objectives during self-supervised training. [12] employed a fully supervised model to generate a depth map as a priori for self-supervised training. It introduces Depth Ranking Loss L_{DR} to regulate the proximity of dynamic objects to the static background, and employs smoothing loss L_N to promote depth map continuity. Notably, executing the supervised network only once at the training onset minimizes additional training and inference costs.

The full supervised network predicts the depth map of the current RGB image, serving as the pseudo-depth truth value during training. M_s calculates geometric inconsistency, dividing the dynamic region. Restricting depth estimation of dynamic regions and enhancing the relationship between far and near points in static regions improve depth prediction for dynamic regions [12]. Model extracts the depth ranking from the pseudo-depth map, constraining network-predicted depth through Depth Ranking Loss L_{DR}:

$$\eta(p_0, p_1) = \log(1 + \exp(-l(p_0 - p_1))), \tag{8}$$

$$L_{DR} = \frac{1}{|\Phi|} \sum_{p \in \Phi} \eta(p), \tag{9}$$

where l represents the ordinal label provided by the depth prior, and Φ represents all sampling point pairs.

To enhance the smoothness of the estimated depth map, Dyna-MSDepth incorporates a calculation of the surface normal by comparing the predicted depth map with the corresponding depth prior. This calculation can be further optimized as follows [12]:

$$L_N = \frac{1}{N}\sum_{i=1}^{N} \left\| n_i - n_i^* \right\|, \tag{10}$$

where n_i represents the surface normal obtained from the predicted depth, while n_i^* represents the normal derived from pseudo-depth. N denotes the total number of pixels in the image.

In the depth prior generated by supervised monocular depth estimation, the edges of the object are segmented very sharp. In order to segment dynamic objects well in the self-supervised monocular depth estimation network, Dyna-Depth further applies the edge normal loss [12], so that the estimated depth map is consistent with the relative normal angle of the edge point pair in the depth prior:

$$L_{EN} = \frac{1}{N}\sum_{i=1}^{N} \left\| n_{Ai} n_{Bi} - n_{Ai}^* n_{Bi}^* \right\|, \tag{11}$$

where n_{Ai} represents the normal of the estimated depth map's point pairs, while n_{Ai}^* represents the normal of the point pairs provided by the depth prior.

Thus, the total loss of Dyna-MSDepth is formulated as:

$$L = \alpha L_G + \beta L_P^M + \gamma L_N + \varphi L_{DR} + \vartheta L_{EN}, \tag{12}$$

where φ and ϑ are weights respectively to punish different losses.

3.3 Multi-scale DepthNet

Current configuration restricts Dyna-MSDepth to fixed-resolution image processing, limiting scale adaptability. Monocular scenes often require capturing objects at varying distances from the camera's optical center, making a single scale insufficient [47]. Employing multi-scale input enhances information coverage for objects of different scales. Additionally, single-scale methods often encounter discontinuities due to crossing large depth disparities [48]. Leveraging multi-scale information smooths depth images, reducing depth value inconsistencies. Furthermore, multi-scale information effectively mitigates image noise [49], artifacts, and enhances depth estimation accuracy and robustness.

Hence, to enhance the capabilities of DepthNet in Dyna-MSDepth, an additional branch is incorporated to handle RGB images with varying resolutions. When the monocular image is inputted to the network, the original resolution image is fed into the existing DepthNet backbone, while a down-sampled low-resolution image is directed to the new branch. Consequently, Dyna-MSDepth exhibits multi-scale characteristics. Moreover, most monocular depth estimation networks utilize ResNet as their backbone, which, although efficient and lightweight, lacks sufficient interaction among high-order spatial features across different levels. This limitation hampers its potential for improved accuracy and robustness. Consequently, this section aims to devise a novel multi-scale DepthNet, integrating the latest gConv model for high-order spatial feature interaction.

The significance of gated convolution (gConv) [50] within Dyna-MSDepth lies in its pivotal role in facilitating multi-scale input and high-order spatial interaction. Assuming that $\boldsymbol{u} \in \mathbb{R}^{H\times W\times C}$ represents the input layer of gConv, the corresponding output feature layer $\boldsymbol{y}$ is mathematically expressed as follows:

$$[\boldsymbol{a}_0^{H\times W\times C}, \boldsymbol{b}_0^{H\times W\times C}] = \phi_{\text{in}}(\boldsymbol{u}) \in \mathbb{R}^{H\times W\times 2C}, \tag{13}$$

$$\boldsymbol{a}_1 = f(\boldsymbol{b}_0) \odot \boldsymbol{a}_0 \in \mathbb{R}^{H\times W\times C}, \tag{14}$$

$$\boldsymbol{y} = \phi_{\text{out}}(\boldsymbol{a}_1) \in \mathbb{R}^{H\times W\times C}, \tag{15}$$

where ϕ_{in} and ϕ_{out} are the linear projection processes, and f represents depth-wise convolution, and $\boldsymbol{a}_0$ and $\boldsymbol{b}_0$ represent the intermediate features in the gConv.

The mathematical description of $\boldsymbol{a}_1$ can be further refined as:

$$a_1^{(i,c)} = \sum\nolimits_{j\in\Psi_i} \omega_{i\to j}^{c} b_0^{(j,c)} a_0^{(i,c)}, \tag{16}$$

where Ψ_i is a local window centered at i, and ω is the weight of depth-wise convolution. By utilizing element-wise multiplication, the interaction between a_0^i and b_0^j at the 1-order spatial level enhances the model's representation capability.

After realizing the 1-order spatial interaction, Dyna-MSDepth further extends it to high-order spatial interaction [50], so that it can learn stronger features. ϕ_{in} is used to further extract high-dimensional features:

$$\begin{aligned}&[\boldsymbol{a}_0^{H\times W\times C_0}, \boldsymbol{b}_0^{H\times W\times C_0}, \ldots, \boldsymbol{b}_{n-1}^{H\times W\times C_{n-1}}] \\ &\quad = \phi_{\text{in}}(\boldsymbol{u}) \in \mathbb{R}^{H\times W\times C_0 + \sum_{0\le k\le n-1} C_k}.\end{aligned} \tag{17}$$

At this stage, the concept of 1-order spatial interaction can be expanded recursively to include higher-order spatial interaction, referred to as g^{n}Conv:

$$\boldsymbol{a}_{k+1} = f_k(\boldsymbol{b}_{k+1}) \odot q_k(\boldsymbol{a}_k)/\alpha, k = 0, 1, \ldots, n-1, \tag{18}$$

where α is a scale factor used to ensure training stability, and f_k represents k depth-wise convolution layers, and q_k is

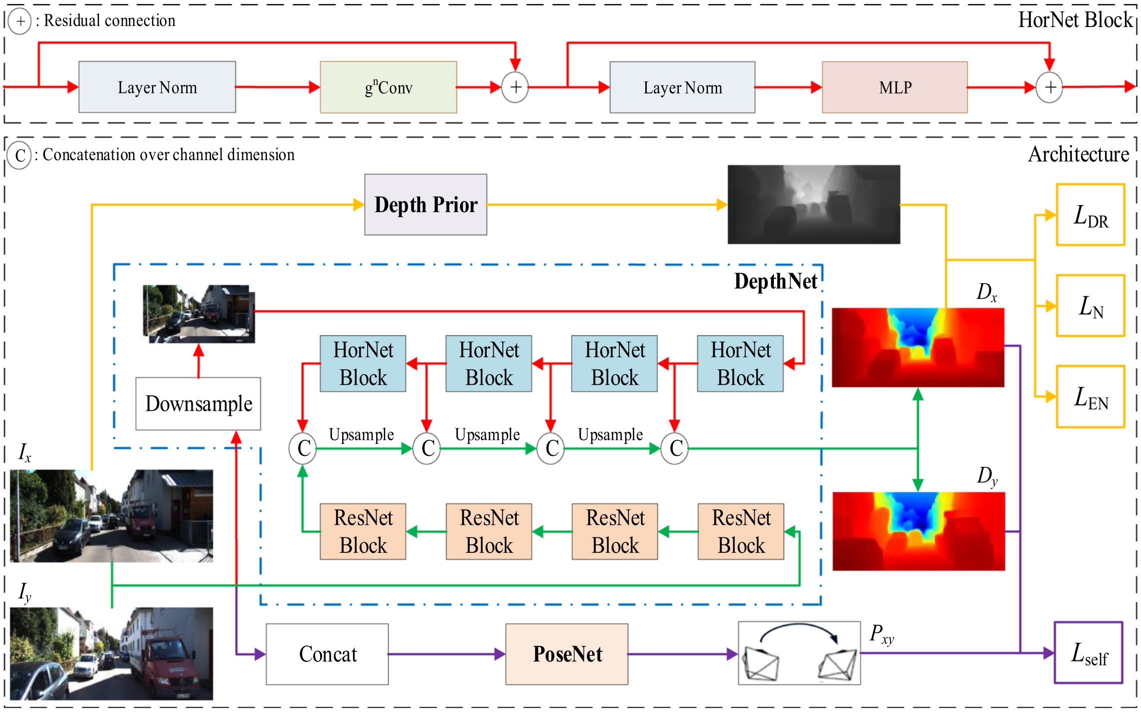

Fig. 2 The architecture of Dyna-MSDepth. It consists of three parts: self-supervised monocular depth estimation, dynamic region optimization, and multi-scale input

utilized for dimension mapping across various feature layers, facilitating dimension matching:

$$q_k = \begin{cases} \text{Identity}, \\ \text{Linear}(C_{k-1}, C_k). \end{cases} \tag{19}$$

The recursive process yields a final output that is passed to ϕ_{out} for output mapping, denoted as g^{n}Conv. Additionally, to reduce computational overhead, feature dimensionality can be optimized as follows:

$$C_k = \frac{C}{2^{n-k-1}}, 0 \leq k \leq n-1. \tag{20}$$

Indeed, the computation of g^{n}Conv entails a coarse-to-fine strategy. Additionally, the computation of g^{n}Conv does not experience a substantial increase as n becomes larger:

$$\text{FLOPs}(\text{g}^n\text{Conv}) < H \times W \times C(2 \times K^2 + 11/3 \times C + 2), \tag{21}$$

where K is the convolution kernel size of depth-wise convolution.

The high-order spatial interaction principle of g^{n}Conv exhibits similarity to the attention mechanism in Transformers. The Transformer architecture, which includes the multi-head self-attention mechanism, has demonstrated promising results [51, 52]. However, it suffers from quadratic complexity in terms of the feature layer size [50]. In contrast, the g^{n}Conv model not only achieves advanced high-order spatial interactions but also reduces computational burden. This reduction is particularly important for successful Monocular depth estimation.

3.4 Model architecture

The architecture of Dyna-MSDepth is illustrated in Fig. 2. The input for Dyna-MSDepth is continuous RGB image frames. When feeding the image pair (I_x, I_y), three data streams are generated. Firstly, a pre-trained supervised network predicts the current frame. The supervised network establishes a stable near-far relationship between dynamic objects and static backgrounds, providing sharp object edges. It should be noted that the supervision network runs only once during the entire training process, minimizing training costs. Secondly, the image pair (I_x, I_y) undergoes depth map extraction through the multi-scale DepthNet, enabling multi-scale, multi-resolution, and high-order spatial interaction feature fusion. Thirdly, PoseNet processes the image pair (I_x, I_y) to determine the relative pose between the images. Using the estimated depth map and relative pose, the loss is computed according to Eq. 12, and gradient backpropagation is applied to complete the training process. Following training, Dyna-MSDepth generates a smooth, stable, sharp-edged, and consistent depth map during the forward inference process. This depth map is qualitatively and quantitatively evaluated, and then utilized in monocular SLAM to mitigate scale drift.

4 Experiment

Dyna-MSDepth, proposed in this paper, aims to estimate reliable multi-scale depth maps in dynamic scenes to restore scale consistency in visual SLAM. This chapter introduces four challenging dynamic datasets (TUM, KITTI, BONN, DDAD) for evaluating the depth estimation performance of Dyna-MSDepth in Sect. 4.1, along with the evaluation metrics in Sect. 4.2. Dyna-MSDepth is trained on these datasets, and the results in Sect. 4.4 demonstrate its superior performance compared to existing leading-edge methods. Furthermore, the estimated depth maps are directly integrated into the visual SLAM system, showcasing their practical applicability through qualitative and quantitative assessments in Sect. 4.5.

4.1 Datasets

KITTI. The KITTI dataset serves as a common benchmark for SLAM, monocular depth estimation, and object detection. Certain sequences in the KITTI dataset pose challenges due to the presence of numerous dynamic objects, leading to potential failures in SLAM feature point tracking and optical flow estimation [7]. Furthermore, monocular depth estimation is adversely affected by the presence of dynamic objects. To mitigate this issue, it becomes necessary to perform monocular depth estimation on the KITTI dataset while accounting for the influence of dynamic objects. In line with previous work [53], 697 images were utilized from the KITTI dataset for testing and the remaining images for training. Image resolution was scaled from 1241×376 to 832×256.

TUM. Similarly, the TUM dataset serves as a crucial benchmark for indoor SLAM systems. Notably, the dynamic sequences in the TUM dataset contain a significant number of dynamic objects that occupy a substantial portion of the image. Conventional SLAM approaches tend to suffer from tracking losses or drift in such dynamic sequences. Therefore, it is imperative to estimate the depth of monocular images in a manner that accounts for the influence of dynamic objects in the TUM dataset. Additionally, the TUM dataset provides depth ground truths, enabling the evaluation of estimated depth maps. For evaluation purposes, the last two dynamic sequences from the TUM dataset were reserved for testing, while the remaining sequences were used for training. Image resolution was scaled from 640×480 to 320×256.

BONN. The BONN dataset comprises 26 indoor sequences featuring fast-moving individuals and other objects. The primary distinction of the BONN dataset lies in the speed of object movement when compared to the TUM dataset. The Pre-set test set was selected for evaluation while employing the remaining sequences for training [12]. Image resolution was scaled to 320×256 to maintain consistency.

DDAD. DDAD, a comprehensive dataset comprising 200 sequences, presents a significant challenge for monocular depth estimation and SLAM due to the predominant presence of moving vehicles. Unlike the KITTI dataset, the DDAD dataset primarily consists of dynamic scenes. The DDAD dataset follows the standard training set/test set segmentation, including 12,650 training images and 3950 test images, all scaled to a resolution of 640×384.

4.2 Evaluation metrics

The evaluation of Dyna-MSDepth encompasses two aspects: monocular depth estimation and SLAM.

For monocular depth estimation, standard evaluation metrics were employed such as mean absolute relative error (AbsRel), root mean squared error (RMS), and accuracy under threshold($\delta_i < 1.25^i, i = 1, 2, 3$). To ensure consistent evaluation, the scale was restored before calculating these metrics. Additionally, this paper used a novel evaluation approach that distinguishes dynamic and static regions based on semantic segmentation [12]. Specifically, this study utilize a pre-trained semantic segmentation network to identify dynamic objects (people and cars) in the four datasets. Subsequently, compute the monocular depth estimation metrics separately for dynamic and static regions.

Regarding the evaluation of the SLAM system, this paper focus on the KITTI datasets. On the KITTI dataset, the primary challenge for monocular SLAM is the scale drift resulting from long sequences. Hence,the extent of scale drift reduction achieved by incorporating the depth map from Dyna-MSDepth estimation was assess, along with the average trajectory accuracy after scale alignment.

4.3 Implementation details

During the training process, the initial learning rate for the four datasets was set to 1e-4 and multiplied by 0.8 every 10 epochs. The batch size for the KITTI and DDAD datasets in outdoor scenes is set to 8, while the batch size for the TUM and BONN datasets in indoor scenes is set to 4. $\alpha = 1$, $\beta = 0.5$, and $\gamma = \varphi = \vartheta = 0.1$. The pre-trained LeReS [54] model was utilized to generate a depth prior and MSeg [55] was utilized for generating dynamic object masks for evaluation.

4.4 Depth estimation results

Figure 3 presents the qualitative comparison results between Dyna-MSDepth and other cutting-edge techniques on the KITTI dataset. The depth map generated by Dyna-MSDepth

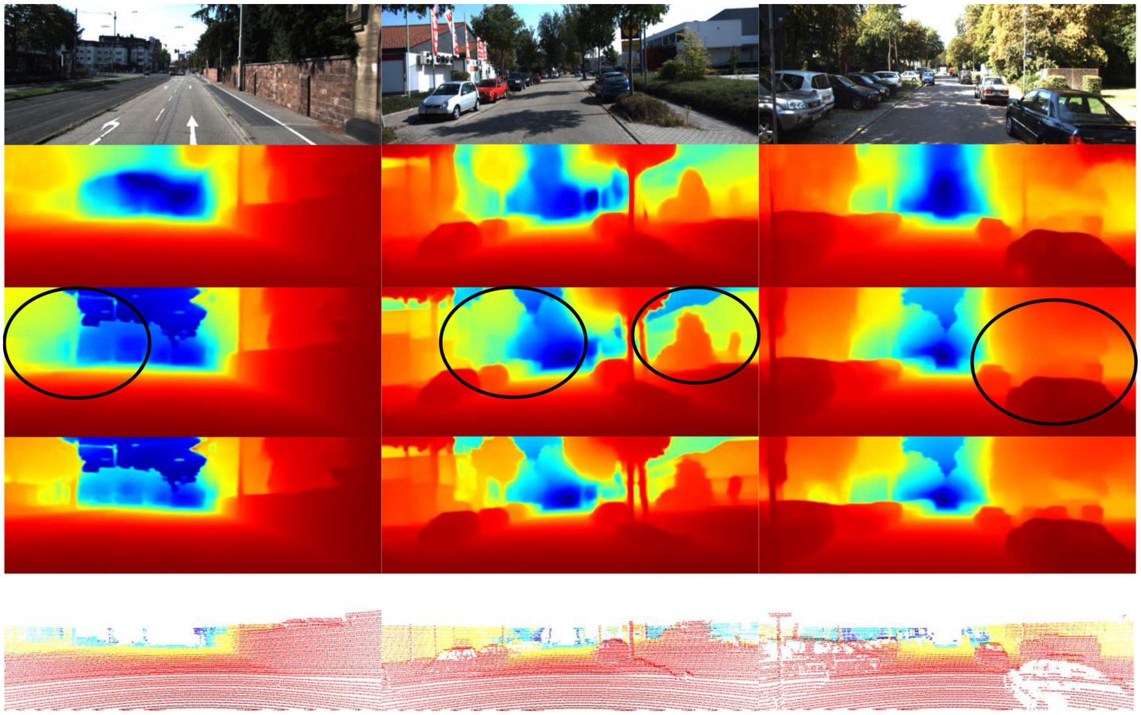

Fig. 3 Qualitative comparison results of Dyna-MSDepth and other SOTA schemes on the KITTI dataset. Top row: the original images from the KITTI dataset. Second row: the monocular depth estimation results of [11]. Third row: the monocular depth estimation results of [12]. Fourth row: the monocular depth estimation results of Dyna-MSDepth. Bottom row: the ground truth depth maps provided by the KITTI dataset

Table 1 Quantitative comparison results of Dyna-MSDepth and other SOTA schemes on the KITTI dataset

Methods	Full image					Dynamic		Static	
	AbsRel	RMS	δ_1	δ_2	δ_3	AbsRel	δ_1	AbsRel	δ_1
Monodepth2	0.114	4.986	0.869	0.956	0.980	0.187	0.731	0.104	0.884
SC-Depth	0.118	4.997	0.860	0.956	0.981	0.242	0.698	0.108	0.878
SC-DepthV3	0.118	4.709	0.864	0.960	0.984	0.205	0.703	0.108	0.881
Dyna-MSDepth	0.111	4.641	0.873	0.965	0.986	0.181	0.733	0.103	0.886

outperforms the previous method. Specifically, the depth estimation for small objects in [11] is highly inaccurate, while for low texture regions in [12], although improved compared to [11], it still falls short of the accuracy achieved by Dyna-MSDepth. Table 1 summarizes the quantitative evaluation, demonstrating the significant performance enhancement of Dyna-MSDepth compared to SC-DepthV3 [12]. Notably, Dyna-MSDepth achieves a 3% improvement in accuracy for dynamic regions. Enhancing performance in small object, low texture, and long distance depth estimation is achieved through the utilization of multi-scale input in Dyna-MSDepth.

Figure 4 presents the qualitative comparison results of Dyna-MSDepth and other state-of-the-art approaches on the TUM test set. The findings demonstrate that depth estimation without a dynamic object loss function yields poor results for dynamic objects. In particular, nearby points fail to estimate accurate depth values, resulting in blurred boundaries and potential SLAM failure. In contrast, SC-DepthV3 [12] exhibits commendable depth estimation for dynamic regions, yet it still experiences missed detection for other close-range dynamic objects (e.g., hands and heads). Additionally, Dyna-MSDepth shows superior performance in capturing low-texture structures at relatively distant distances, showcasing the benefits of multi-scale input. The quantitative indicators in Table 2 also verify this conclusion.

Figure 5 illustrates the qualitative comparison between Dyna-MSDepth and other leading-edge methods on the BONN dataset. The findings reveal that without optimizing the dynamic region loss function, monocular depth estimation results of Dyna-MSDepth for dynamic targets are notably poor, with the depth value of dynamic objects closely

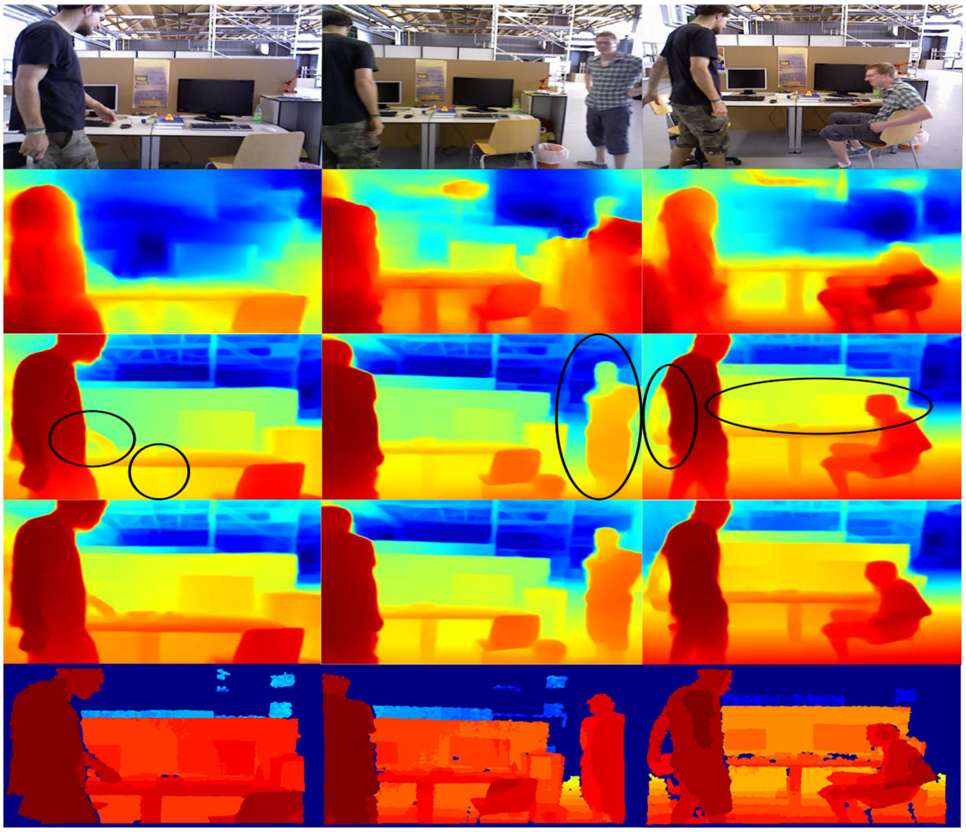

Fig. 4 Qualitative comparison results of Dyna-MSDepth and other SOTA schemes on the TUM dataset. Top row: the original images from the TUM dataset. Second row: the monocular depth estimation results of [13]. Third row: the monocular depth estimation results of [12]. Fourth row: The monocular depth estimation results of Dyna-MSDepth. Bottom row: the ground truth depth maps provided by the TUM dataset

Table 2 Quantitative comparison results of Dyna-MSDepth and other cutting-edge methods on the TUM dataset

Methods	Full image					Dynamic		Static	
	AbsRel	RMS	δ_1	δ_2	δ_3	AbsRel	δ_1	AbsRel	δ_1
Monodepth2	0.312	1.408	0.474	0.793	0.905	0.431	0.348	0.262	0.526
SC-Depth	0.257	0.283	0.616	0.814	0.909	0.512	0.274	0.176	0.715
SC-DepthV2	0.223	0.282	0.643	0.862	0.932	0.283	0.494	0.206	0.686
SC-DepthV3	0.163	0.265	0.797	0.882	0.937	0.165	0.796	0.171	0.780
Dyna-MSDepth	0157	0.259	0.801	0.885	0.947	0.162	0.802	0.161	0.783

linked to the static background. Additionally, due to the lack of multi-scale characteristics, the depth estimation results for the static background are subpar. In contrast, the current SOTA method, SC-DepthV3 [12], exhibits significant improvement in deep optimization for dynamic regions. Nonetheless, there are still limitations in SC-DepthV3's depth estimation results when dynamic objects enter the scene or when two moving objects overlap, making the extraction of object edges less distinct. Conversely, Dyna-MSDepth demonstrates superior performance in handling dynamic objects, and its multi-scale input enables precise capturing of intricate details of distant objects like tables and chairs. Table 3 presents the quantitative comparison results of Dyna-MSDepth on the BONN dataset, indicating its overall superiority over SC-DepthV3, particularly in terms of dynamic region accuracy, which exhibits a 1.2% enhancement.

Figure 6 illustrates the qualitative comparison results of Dyna-MSDepth and other state-of-the-art techniques on the DDAD dataset. The depth map produced by method [11] exhibits poor discrimination between near and far points, resulting in inaccurate depth estimation for dynamic objects. On the other hand, SC-DepthV3 yields sharp edges for dynamic objects but struggles with accurate depth estimation for distant low-texture scenes and objects. In contrast, leveraging multi-scale input, Dyna-MSDepth demonstrates improved depth estimation for small objects in the distance. The comparison results on the DDAD dataset (Table 4) reveal that Dyna-MSDepth, discussed in this study, exhibits slightly lower accuracy in dynamic regions compared to SC-DepthV3, albeit with similar processing methods. However,

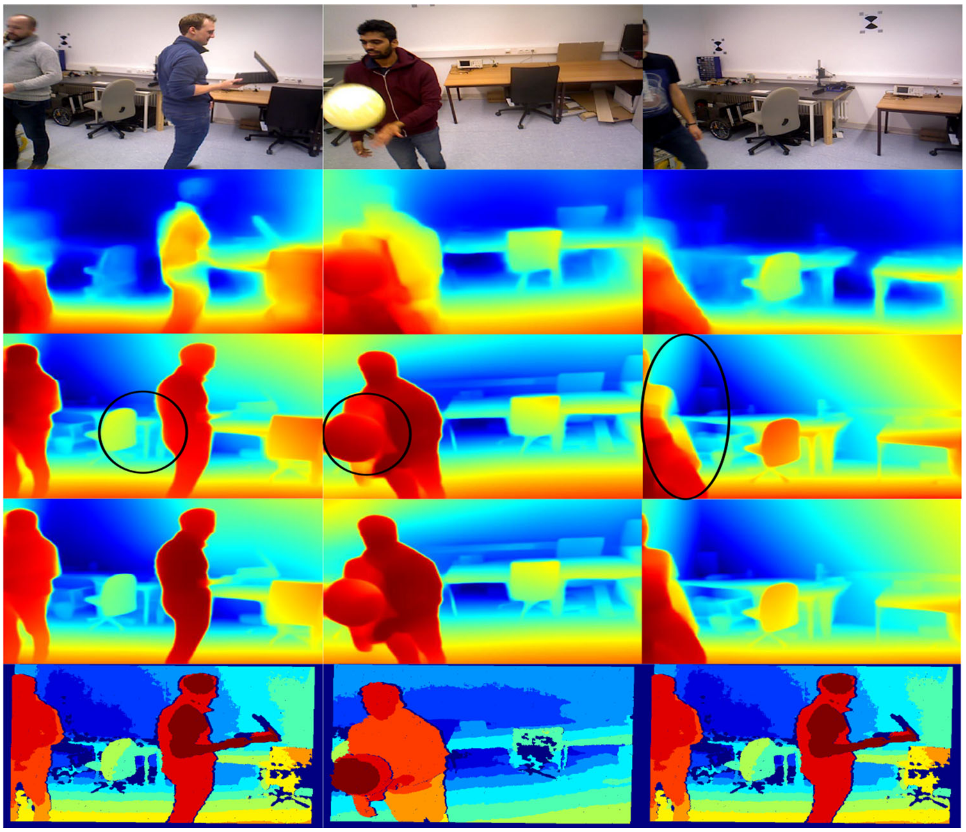

Fig. 5 Qualitative comparison results of Dyna-MSDepth and other SOTA schemes on the BONN dataset. Top row: the original images from the BONN dataset. Second row: the monocular depth estimation results of [13]. Third row: the monocular depth estimation results of [12]. Fourth row: the monocular depth estimation results of Dyna-MSDepth. Bottom row: the ground truth depth maps provided by the BONN dataset

Table 3 Quantitative comparison results of Dyna-MSDepth and other leading-edge methods on the BONN dataset

Methods	Full image					Dynamic		Static	
	AbsRel	RMS	δ_1	δ_2	δ_3	AbsRel	δ_1	AbsRel	δ_1
Monodepth2	0.565	2.337	0.352	0.591	0.728	0.474	0.172	0.594	0.383
SC-Depth	0.272	0.733	0.623	0.858	0.948	0.704	0.166	0.180	0.714
SC-DepthV2	0.211	0.619	0.714	0.873	0.936	0.488	0.247	0.152	0.803
SC-DepthV3	0.126	0.379	0.889	0.961	0.980	0.220	0.720	0.102	0.931
Dyna-MSDepth	0.120	0.385	0.898	0.967	0.984	0.193	0.732	0.104	0.929

Dyna-MSDepth's multiscale architecture notably enhances its ability to capture detailed textures and distant depth, leading to superior overall performance when compared to SC-DepthV3.

4.5 SLAM test results

In outdoor settings like autonomous driving, SLAM systems require prolonged operation, exacerbating scale and trajectory drift issues. To assess Dyna-MSDepth's efficacy in mitigating these challenges, this study conducts experiments with ORB-SLAM3 on the KITTI dataset in monocular and RGB-D modes, evaluating positioning accuracy and scale drift. Loop detection is disabled in both experiments to better gauge the depth map's impact on scale restoration.

The KITTI dataset, commonly utilized for SLAM experiments in outdoor environments, contains sequences with a substantial presence of dynamic objects. The Dyna-MSDepth generated depth map is directly employed in ORB-SLAM3, followed by trajectory evaluation for each sequence. The Fig. 7 demonstrates the pronounced scale drift occurring due to the monocular camera's continuous operation over vast distances. With the integration of the depth map, a substantial alleviation of scale drift is observed. This improvement in scale accuracy is further validated by the data presented in the Table 5, highlighting the significant enhancement in positioning precision achieved through the incorporation of Dyna-MSDepth generated depth maps.

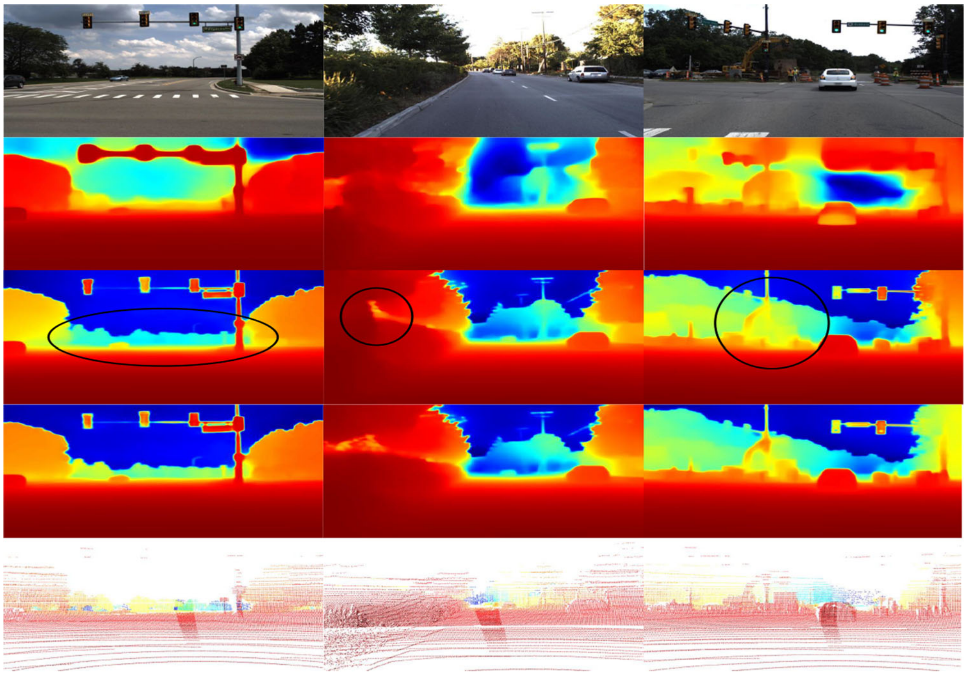

Fig. 6 Qualitative comparison results of Dyna-MSDepth and other SOTA schemes on the DDAD dataset. Top row: the original images from the DDAD dataset. Second row: the monocular depth estimation results of [11]. Third row: the monocular depth estimation results of [12]. Fourth row: the monocular depth estimation results of Dyna-MSDepth. Bottom row: the ground truth depth maps provided by the DDAD dataset

Table 4 Quantitative comparison results of Dyna-MSDepth and other cutting-edge methods on the DDAD dataset

Methods	Full image					Dynamic		Static	
	AbsRel	RMS	δ_1	δ_2	δ_3	AbsRel	δ_1	AbsRel	δ_1
Monodepth2	0.239	18.392	0.752	0.899	0.949	0.747	0.432	0.188	0.771
SC-Depth	0.169	16.290	0.773	0.905	0.951	0.345	0.546	0.155	0.783
PackNet	0.182	15.021	0.828	0.925	0.961	0.564	0.520	0.137	0.843
SC-DepthV3	0.142	15.868	0.813	0.922	0.963	0.199	0.697	0.140	0.813
Dyna-MSDepth	0.131	14.650	0.833	0.933	0.971	0.201	0.696	0.127	0.837

5 Conclusion

Monocular SLAM is widely used in visual localization, mapping, 3D reconstruction, and navigation tasks due to its low cost and easy configuration. However, it suffers from scale ambiguity and significant scale drift during long-term running, particularly in dynamic scenes where geometric consistency assumptions are violated.

To address these issues, this paper proposes Dyna-MSDepth, a self-supervised monocular depth estimation network that restores scale consistency and provides globally consistent depth maps in dynamic scenes. Dyna-MSDepth employs self-supervised training and a specific loss function to generate dense depth maps with continuous values, ensuring scale consistency for monocular SLAM. A dynamic optimization strategy is introduced to estimate reliable depth maps in the presence of dynamic objects. Furthermore, multi-scale inputs are introduced to enable Dyna-MSDepth to perceive the depth values of objects with different distances and scales.

Qualitative and quantitative evaluations on challenging dynamic datasets (KITTI, TUM, BONN, DDAD) demonstrate that Dyna-MSDepth outperforms existing state-of-the-art methods in monocular depth estimation. Monocular SLAM experiments on the KITTI datasets further confirm the effectiveness of Dyna-MSDepth in enabling accurate mapping and navigation tasks.

The paper concludes with the following findings:

1. The proposed Dyna-MSDepth can estimate stable, reliable and consistent multi-scale depth maps in dynamic scenes;
2. Evaluation on four challenging dynamic datasets demonstrates that Dyna-MSDepth outperforms other state-of-the-art methods, as observed through qualitative and quantitative analysis;
3. The depth maps generated by Dyna-MSDepth can be directly utilized in monocular SLAM without the need for additional complex post-processing, highlighting its practical applicability.

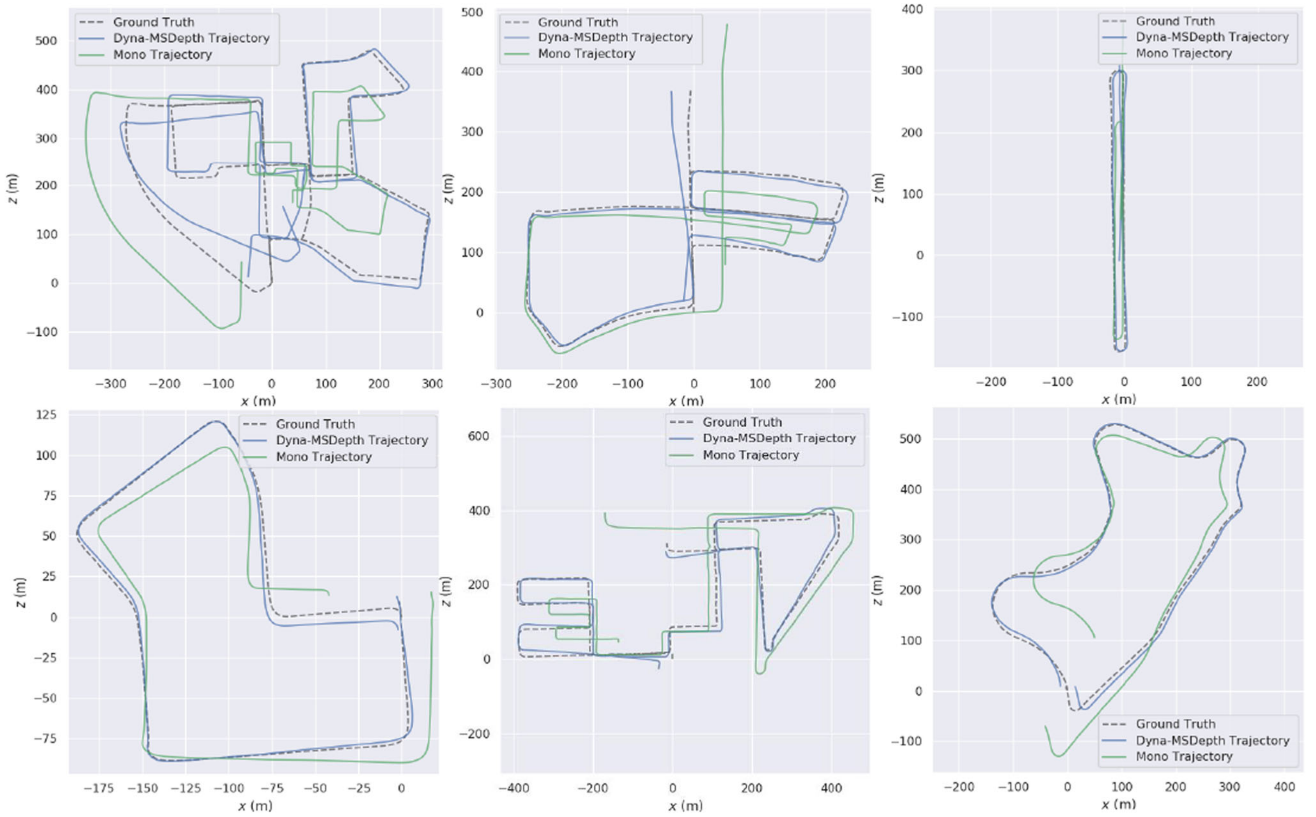

Fig. 7 Qualitative comparison results show that the depth map estimated by Dyna-MSDepth is directly used to reduce the scale drift of monocular SLAM. The sequence from left to right is 00, 05, 06, 07, 08, 09

Table 5 The quantitative comparison results (m) of Dyna-MSDepth before and after the introduction of the depth map estimated on the KITTI dataset

Method	00	01	02	03	04	05	06	07	08	09	10
Monocular	75.6501	275.463	44.5924	2.29264	2.39428	38.2833	42.9787	14.1814	47.9749	50.3000	10.8759
RGBD w/ Dyna-MSDepth	22.6451	112.219	18.5262	0.715301	1.12295	9.70304	5.01295	3.27497	12.4913	7.79669	6.83360

Author Contributions The project design and paper writing were carried out by all authors. JY conceived the core ideas and principles of the paper. YL designed the experiments, trained the models, and wrote the paper. JL collected the data. All authors read and approved the final manuscript.

Funding The authors did not receive support from any organization for the submitted work.

Code Availability Code is available at https://github.com/PepperFlavoredChewingGum/Dyna-MSDepth.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Informed consent The research did not involves human participants or animals.

References

1. Guillaume, T., Evangeline, P., Benazouz, B., et al.: On line mapping and global positioning for autonomous driving in urban environment based on evidential SLAM. In: Paper Presented at the IEEE Intelligent Vehicles Symposium, Seoul, South Korea, 28 June–1 July (2015). https://doi.org/10.1109/IVS.2015.7225785
2. Mostafa, E., Rongjun, Q.: Cross-view slam solver: global pose estimation of monocular ground-level video frames for 3d reconstruction using a reference 3d model from satellite images. ISPRS J. Photogramm. Remote. Sens. **188**(6), 62–74 (2022). https://doi.org/10.1016/j.isprsjprs.2022.03.018

3. Kumar, R.S., Singh, C.D., Ziad, A.-H., et al.: Poni: potential functions for objectgoal navigation with interaction-free learning. In: Paper Presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, USA, 19–24 June (2022). https://doi.org/10.1109/CVPR52688.2022.01832
4. Qi, L., Yue, W., Yilun, W., et al.: Hdmapnet: an online hd map construction and evaluation framework. In: Paper Presented at the International Conference on Robotics and Automation, Philadelphia, USA, 23–27 May (2022). https://doi.org/10.1109/icra46639.2022.9812383
5. Georges, Y., Daniel, A., Elie, S., et al.: Keyframe-based monocular slam: design, survey, and future directions. Robot. Auton. Syst. **98**(12), 67–88 (2017). https://doi.org/10.1016/j.robot.2017.09.010
6. Hanwei, Z., Hideaki, U., Shintaro, O., et al.: MOTSLAM: MOT-assisted monocular dynamic SLAM using single-view depth estimation. In: Paper Presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyoto, Japan, 23–17 October (2022). https://doi.org/10.1109/IROS47612.2022.9982280
7. Carlos, C., Richard, E., Gómez, R.J.J., et al.: Orb-slam3: an accurate open-source library for visual, visual-inertial, and multimap slam. IEEE Trans. Robot. **37**(6), 1874–1890 (2021). https://doi.org/10.1109/TRO.2021.3075644
8. Riccardo, G., Wolfgang, S., Armin, W., et al.: Challenges of slam in extremely unstructured environments: the DLR planetary stereo, solid-state lidar, inertial dataset. IEEE Robot. Autom. Lett. **7**(4), 8721–8728 (2022). https://doi.org/10.1109/LRA.2022.3188118
9. Xiaoyang, L., Liang, L., Mengmeng, W., et al.: Hr-depth: high resolution self-supervised monocular depth estimation. In: Paper Presented at the AAAI Conference on Artificial Intelligence, Vancouver, Canada, 2–9 February (2021). https://doi.org/10.1609/aaai.v35i3.16329
10. Clément, G., Oisin, M.A., Michael, F., et al.: Digging into self-supervised monocular depth estimation. In: Paper Presented at the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November (2019). https://doi.org/10.1109/ICCV.2019.00393
11. JiaWang, B., Huangying, Z., Naiyan, W., et al.: Unsupervised scale-consistent depth learning from video. Int. J. Comput. Vis. **129**(9), 2548–2564 (2021). https://doi.org/10.1007/s11263-021-01484-6
12. Sun, L., Bian, J., Zhan, H., et al.: Sc-depthv3: robust self-supervised monocular depth estimation for dynamic scenes. IEEE Trans. Pattern Anal. Mach. Intell. **46**(1), 497–508 (2023). https://doi.org/10.1109/TPAMI.2023.3322549
13. JiaWang, B., Huangying, Z., Naiyan, W., et al.: Auto-rectify network for unsupervised indoor depth estimation. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 9802–9813 (2021). https://doi.org/10.1109/TPAMI.2021.3136220
14. Ruben, G.-O., Francisco-Angel, M., David, Z.-N., et al.: Pl-slam: a stereo slam system through the combination of points and line segments. IEEE Trans. Robot. **35**(3), 734–746 (2019). https://doi.org/10.1109/TRO.2019.2899783
15. Sturm, J., Nikolas, E., Felix, E., et al.: A benchmark for the evaluation of RGB-D SLAM systems. In: Paper Presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, Portugal, 12–15 October (2012). https://doi.org/10.1109/IROS.2012.6385773
16. Jun, Y., Dongting, L., Fei, Y., et al.: A novel lidar-assisted monocular visual slam framework for mobile robots in outdoor environments. IEEE Trans. Instrum. Meas. **71**(6), 1–11 (2022). https://doi.org/10.1109/TIM.2022.3190031
17. Ayush, K., Shrinivas, P., Eli, P., et al.: Comparison of visual SLAM and IMU in tracking head movement outdoors. Behav. Res. Methods **7**(2), 1–13 (2022). https://doi.org/10.3758/s13428-022-01941-1
18. Luke, T.J., Lam, P.S., Abdesselam, B.: D-net: a generalised and optimised deep network for monocular depth estimation. IEEE Access **9**(8), 134543–134555 (2021). https://doi.org/10.1109/ACCESS.2021.3116380
19. Raul, M.-A., Tardos, J.D.: Orb-slam2: an open-source slam system for monocular, stereo, and RGB-D cameras. IEEE Trans. Robot. **33**(5), 1255–1262 (2017). https://doi.org/10.1109/TRO.2017.2705103
20. Huangying, Z., Saroj, W.C., Jia-Wang, B., et al.: Visual odometry revisited: what should be learnt? In: Paper Presented at the IEEE International Conference on Robotics and Automation, Xian, China, 31 May–5 June (2020). https://doi.org/10.1109/ICRA40945.2020.9197374
21. Dingfu, Z., Yuchao, D., Hongdong, L.: Reliable scale estimation and correction for monocular visual odometry. In: Paper Presented at the IEEE Intelligent Vehicles Symposium, Gothenburg, Sweden, 19–22 June (2016). https://doi.org/10.1109/IVS.2016.7535431
22. Françani, A.O., Maximo, M.R.O.A.: Dense prediction transformer for scale estimation in monocular visual odometry. In: Paper Presented at the Latin American Robotics Symposium, São Bernardo do Campo, Brazil, 18–21 October (2022). https://doi.org/10.1109/LARS/SBR/WRE56824.2022.9995735
23. Danpeng, C., Shuai, W., Weijian, X., et al.: VIP-SLAM: an efficient tightly-coupled RGB-D visual inertial planar SLAM. In: Paper Presented at the IEEE International Conference on Robotics and Automation, Philadelphia, USA, 23–27 May (2022). https://doi.org/10.1109/ICRA46639.2022.9812354
24. Wei, Y., Yifan, L., Chunhua, S., et al.: Enforcing geometric constraints of virtual normal for depth prediction. In: Paper Presented at the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–02 November (2019). https://doi.org/10.1109/ICCV.2019.00578
25. Lam, H., Phong, N.-H., Jiri, M., et al.: Guiding monocular depth estimation using depth-attention volume. In: Paper Presented at the European Conference on Computer Vision, Glasgow, US, 23–27 August (2020). https://doi.org/10.1007/978-3-030-58574-7_35
26. Matteo, P., Filippo, A., Fabio, T., et al.: On the uncertainty of self-supervised monocular depth estimation. In: Paper Presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, USA, 13–19 June (2020). https://doi.org/10.1109/CVPR42600.2020.00329
27. Marvin, K., Jan-Aike, T., Jonas, M., et al.: Self-supervised monocular depth estimation: solving the dynamic object problem by semantic guidance. In: Paper Presented at the European Conference on Computer Vision, Glasgow, US, 23–27 August (2020). https://doi.org/10.1007/978-3-030-58565-5_35
28. Cheng, Z., James Chenhao Liang, G.T., et al.: Adversarial training of self-supervised monocular depth estimation against physical-world attacks. In: Paper Presented at the Eleventh International Conference on Learning Representations, Kigali, Rwanda, 01–05 May (2023). https://doi.org/10.48550/arXiv.2301.13487
29. Cheng, Z., James Liang, H.C., et al.: Physical attack on monocular depth estimation with optimal adversarial patches. In: Paper Presented at the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October (2022). https://doi.org/10.1007/978-3-031-19839-7_30
30. Cheng, Z., Hongjun Choi, J.L., et al.: Fusion is not enough: single modal attacks on fusion models for 3D object detection. In: Paper Presented at the Eleventh International Conference on Learning Representations, Vienna, Austria, 07–11 May (2024). https://doi.org/10.48550/arXiv.2304.14614
31. Chao, Y., Zuxin, L., Xin-Jun, L., et al.: DS-SLAM: a semantic visual SLAM towards dynamic environments. In: Paper Presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, 01–05 October (2018). https://doi.org/10.1109/IROS.2018.8593691

32. Berta, B., Fácil, J.M., Javier, C., et al.: Dynaslam: tracking, mapping, and inpainting in dynamic scenes. IEEE Robot. Autom. Lett. **3**(4), 4076–4083 (2018). https://doi.org/10.1109/LRA.2018.2860039
33. Linyan, C., Chaowei, M.: Sdf-slam: semantic depth filter slam for dynamic environments. IEEE Access **8**(1), 95301–95311 (2020). https://doi.org/10.1109/ACCESS.2020.2994348
34. Jianheng, L., Xuanfu, L., Yueqian, L., et al.: RGB-D inertial odometry for a resource-restricted robot in dynamic environments. IEEE Robot. Autom. Lett. **7**(4), 9573–9580 (2022). https://doi.org/10.1109/LRA.2022.3191193
35. Shihao, S., Yilin, C., Wenshan, W., et al.: DytanVO: joint refinement of visual odometry and motion segmentation in dynamic environments. In: Paper Presented at the IEEE International Conference on Robotics and Automation, London, United Kingdom, 29 May–02 June (2023). https://doi.org/10.1109/ICRA48891.2023.10161306
36. Berta, B., Carlos, C., Tardós, J.D., et al.: Dynaslam II: tightly-coupled multi-object tracking and slam. IEEE Robot. Autom. Lett. **6**(3), 5191–5198 (2021). https://doi.org/10.1109/LRA.2021.3068640
37. Yanwei, P., Tiancai, W., Muhammad, A.R., et al.: Efficient featurized image pyramid network for single shot detector. In: Paper Presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, USA, 15–20 June (2019). https://doi.org/10.1109/CVPR.2019.00751
38. Gouthamaan, M., Swaminathan, J.: Focal-WNet: an architecture unifying convolution and attention for depth estimation. In: Paper Presented at the IEEE 7th International conference for Convergence in Technology, Mumbai, India, 07–09 April (2022). https://doi.org/10.1109/I2CT54291.2022.9824488
39. Junjie, K., Qifei, W., Yilin, W., et al.: Musiq: multi-scale image quality transformer. In: Paper Presented at the IEEE/CVF International Conference on Computer Vision, Montreal, Canada, 10–17 October (2021). https://doi.org/10.1109/ICCV48922.2021.00510
40. Lina, Y., Fengqi, Z., Shen-Pei, W.P., et al.: Multi-scale spatial-spectral fusion based on multi-input fusion calculation and coordinate attention for hyperspectral image classification. Pattern Recogn. **122**(8), 1–13 (2022). https://doi.org/10.1016/j.patcog.2021.108348
41. Peng, L., Tran, T.C., Bin, K., et al.: Cada: multi-scale collaborative adversarial domain adaptation for unsupervised optic disc and cup segmentation. Neurocomputing **469**(2), 209–220 (2022). https://doi.org/10.1016/j.neucom.2021.10.076
42. Kumar, J.A., Rajeev, S.: Detection of copy-move forgery in digital image using multi-scale, multi-stage deep learning model. Neural Process. Lett. **51**(12), 75–100 (2022). https://doi.org/10.1007/s11063-021-10620-9
43. Xinxin, Z., Long, Z.: Sa-fpn: an effective feature pyramid network for crowded human detection. Appl. Intell. **52**(6), 12556–12568 (2022). https://doi.org/10.1007/s10489-021-03121-8
44. Yuancheng, L., Shenglong, Z., Hui, C.: Attention-based fusion factor in fpn for object detection. Appl. Intell. **52**(8), 15547–15556 (2022). https://doi.org/10.1007/s10489-022-03220-0
45. Ravi, G., Kumar, B.V., Gustavo, C., et al.: Unsupervised cnn for single view depth estimation: geometry to the rescue. In: Paper Presented at the European Conference on Computer Vision, Amsterdam, Netherlands, 10–16 October (2016). https://doi.org/10.1007/978-3-319-46484-8_45
46. Tinghui, Z., Matthew, B., Noah, S., et al.: Unsupervised learning of depth and ego-motion from video. In: Paper Presented at the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, 21–26 July (2017). https://doi.org/10.1109/CVPR.2017.700
47. Zige, W., Zhen, C., Congxuan, Z., et al.: Lcif-net: local criss-cross attention based optical flow method using multi-scale image features and feature pyramid. Signal Process. Image Commun. **112**(14), 1–13 (2023). https://doi.org/10.1016/j.image.2023.116921
48. Dong, N., Rui, L., Ling, W., et al.: Pyramid architecture for multi-scale processing in point cloud segmentation. In: Paper Presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, USA, 18–24 June (2022). https://doi.org/10.1109/CVPR52688.2022.01677
49. Kalyan, S., Johnson, M.K., Wojciech, M., et al.: Multi-scale image harmonization. ACM Trans. Graph. **29**(4), 1–10 (2010). https://doi.org/10.1145/1778765.1778862
50. Yongming, R., Wenliang, Z., Yansong, T., et al.: Hornet: efficient high-order spatial interactions with recursive gated convolutions. Adv. Neural Inf. Process. Syst. **35**(4), 10353–10366 (2022). https://doi.org/10.48550/arXiv.2207.14284
51. Sanghyun, W., Shoubhik, D., Ronghang, H., et al.: Convnext v2: co-designing and scaling convnets with masked autoencoders, pp. 1–16 (2023) arXiv:2301.00808. https://doi.org/10.48550/arXiv.2301.00808
52. Ding, X., Zhang, X., Zhou, Y., et al.: Scaling up your kernels to 31 × 31: revisiting large kernel design in CNNs. In: Paper Presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, USA, 18–24 June (2022). https://doi.org/10.1109/CVPR52688.2022.01166
53. Clément, G., Oisin, M.A., Michael, F., et al.: Digging into self-supervised monocular depth estimation. In: Paper Presented at the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–02 November (2019). https://doi.org/10.1109/ICCV.2019.00393
54. Wei, Y., Jianming, Z., Oliver, W., et al.: Learning to recover 3d scene shape from a single image. In: Paper Presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, USA, 20–25 June (2021). https://doi.org/10.1109/CVPR46437.2021.00027
55. John, L., Zhuang, L., Ozan, S., et al.: MSeg: a composite dataset for multi-domain semantic segmentation. In: Paper Presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, USA, 13–19 June (2020). https://doi.org/10.1109/CVPR42600.2020.00295

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Jianjun Yao received a Ph.D. degree from Harbin Institute of Technology, China. He is a professor of Mechanical Engineering at Harbin Engineering University, China. His research interests include mechanical transmission and control, servo mechanism, robot technology, intelligent manufacturing equipment, environmental simulation testing technology, intelligent healthcare equipment, etc.

Yingzhao Li received a master degree from Harbin Engineering University, China. His research activities cover machine vision, autonomous driving, artificial intelligence, etc.

Jiajia Li is currently finishing a master degree at College of Mechanical and Electrical Engineering, Harbin Engineering University, China. Her research activities cover robot technology, mechatronics, vibration and noise reduction, etc.