Online Mutual Adaptation of Deep Depth Prediction and Visual SLAM

Shing Yan Loo¹,², Moein Shakeri¹, Sai Hong Tang², Syamsiah Mashohor² and Hong Zhang¹

Abstract—The ability of accurate depth prediction by a convolutional neural network (CNN) is a major challenge for its wide use in practical visual simultaneous localization and mapping (SLAM) applications, such as enhanced camera tracking and dense mapping. This paper is set out to answer the following question: Can we tune a depth prediction CNN with the help of a visual SLAM algorithm even if the CNN is not trained for the current operating environment in order to benefit the SLAM performance? To this end, we propose a novel online adaptation framework consisting of two complementary processes: a SLAM algorithm that is used to generate keyframes to fine-tune the depth prediction and another algorithm that uses the online adapted depth to improve map quality. Once the potential noisy map points are removed, we perform global photometric bundle adjustment (BA) to improve the overall SLAM performance. Experimental results on both benchmark datasets and a real robot in our own experimental environments show that our proposed method improves the overall SLAM accuracy. While regularization has been shown to be effective in multi-task classification problems, we present experimental results and an ablation study to show the effectiveness of regularization in preventing catastrophic forgetting in the online adaptation of depth prediction, a single-task regression problem. In addition, we compare our online adaptation framework against the state-of-the-art pre-trained depth prediction CNNs to show that our online adapted depth prediction CNN outperforms the depth prediction CNNs that have been trained on a large collection of datasets.

I. INTRODUCTION

Long-term adaptation in robotics is challenging yet invaluable for many practical applications, ranging from safe autonomous driving to human-robot interaction. This paper is concerned with two related but distinct problems: visual SLAM and single-image depth prediction. Visual SLAM is a problem concerning estimating the structure and camera motion, whereby the camera is attached to a mobile robot or device. On the other hand, single-image depth prediction is a method of predicting a depth map from an image. In particular, we investigate the problem of visual SLAM with online adaptation by on-demand fine-tuning of a depth prediction convolutional neural network (CNN) and integrating online adapted depth prediction to improve structure and motion estimation.

Given a pre-trained depth prediction CNN—which may not necessarily perform well in a real-world setting—can we still integrate the CNN in a traditional SLAM pipeline such that the depth prediction CNN and the SLAM performance are simultaneously improved? To address this problem, we propose an on-demand adaptation framework. That is, we perform SLAM in an environment whose keyframes are used to determine the quality of predicted depth and, depending on the quality, we switch between fine-tuning of the depth prediction CNN and incorporating fine-tuned depth prediction into global BA to improve SLAM accuracy. The main contributions of our work are as follows:

• We propose a novel online adaptation framework to fine-tune monocular depth prediction on-demand and perform global BA to improve SLAM accuracy. Experimental results in real-world settings demonstrate an improvement in SLAM accuracy after incorporating online adapted depth in global photometric BA.

• To perform online adaptation without catastrophic forgetting, we investigate the effectiveness of EWC regularization in the online adaptation of a depth prediction CNN, a single-task regression problem, as opposed to heavily studied regularization in multi-task classification problems [1], [2]. Experimental results show that use of CNN parameter importance regularization retains the previously learned knowledge better than using only experience replay [3]–[5].

• To improve the SLAM accuracy using the adapted depth prediction CNN, we propose using online adapted depth for culling the potential noisy map points before performing global photometric BA, resulting in an improvement in camera tracking and map reconstruction.

II. RELATED WORK

A modern SLAM algorithm [6] typically consists of a front-end and a back-end. Particularly in visual SLAM, the front-end extracts and matches reliable and repeatable features in the images captured by a camera. Two dominant approaches used in the front-end are indirect [7], [8] and direct methods [9], [10] whose goal is to define the observation, odometry and optionally loop constraints for the back-end optimization. Indirect methods preprocess images to generate intermediate feature representation such as handcrafted ORB [11] and deep CNN descriptors [12]–[14] to facilitate feature matching. Alternatively, direct methods operate directly on the raw sensory data—the image pixels—for establishing constraints. Then, the back-end performs a maximum a posteriori (MAP), e.g., BA [15] and pose-graph optimization [16], on the established...
constraints to obtain optimized SLAM variables consisting of a set of camera poses and the position of 3D points obtained from the matched visual features.

Relevantly, we also review background literature on single-image depth prediction. Predicting depth from a single image has shown tremendous progress recently with CNNs. The three main pillars that drive depth prediction accuracy are improved training methods, novel CNN architectures and rich and diverse training datasets. One of the pioneering works is the supervised learning of global and local depth information with a CNN proposed by Eigen et al., which requires ground truth labels to train the CNN parameters. Another seminal research direction is unsupervised learning using photometric reconstruction loss from binocular [17], [18] and motion stereo [19] to overcome the need for ground truth depth. For improved training accuracy, semi-supervised learning incorporates photometric reconstruction and ground truth depth losses [20]. Recently, novel CNN architectures have been proposed to improve depth prediction accuracy, including discretization of the depth output [21], [22], 3D packing and unpacking convolutional blocks [23], dense prediction transformers [24] and generative adversarial networks [25]. Furthermore, single-image relative depth prediction, such as MiDaS [26] and DiverseDepth [27], has been shown to predict accurate relative depth on diverse scene types by leveraging an extensive collection of training datasets.

To overcome the domain gap of single-image depth prediction, online learning has been used to fine-tune a depth prediction CNN on the images available in the target domain. These images typically arrive sequentially [3], [4], [28]. One particular challenge for online fine-tuning is ensuring that learning does not overfit the most recent data, a problem known as catastrophic forgetting [29]. Experience replay has been proposed to mitigate catastrophic forgetting by inserting randomly sampled past training data into the current training batch [3]–[5]. An alternative solution is to regularize and preserve the CNN parameters that are important to the previously learned tasks [29], [30]. In a typical multi-task learning scenario, the importance of the CNN parameters can be measured by the magnitude of the gradients with respect to the loss function or output function through the training on a task [30]–[33], which intuitively determines how a perturbation in a CNN parameter affects the loss. Instead of measuring the parameter importance after learning a task and regularizing the parameters in learning the next task, Maltoni and Lomonaco [33] propose single incremental-task (SIT) learning, which seeks to estimate and consolidate the parameter importance from batch to batch using synaptic intelligence (SI), so that the previously learned knowledge is retained throughout the learning. Unlike SIT, which has to deal with learning new instances or classes, we consider single-task learning whose goal is to preserve the previously learned depth from batch to batch by consolidating the parameter importance using a semi-supervised loss, instead of solving classification problems with ground truth labels [33].

Depth prediction by a CNN has been widely used to benefit SLAM algorithms. A SLAM pipeline can be enhanced by improving feature matching [34] and photometric re-projection accuracies [35] through the use of depth prediction by CNNs. However, the predicted depth has to be accurate to achieve good performance, an assumption that is often violated in practical robotics applications. Therefore, instead of directly incorporating the predicted depth information into front-end tracking and mapping [34], [35], we design a VO pipeline that only incorporates online adapted depth information in optimizing the structure and motion estimation. To this end, we use two common procedures in SLAM optimization: map point culling and global BA. Removing noisy map points is an essential part of sparse SLAM for preventing erroneous state estimation from being included in optimization (see ORB-SLAM3 [8] and DSO [9]; on the other hand, global photometric BA [36], [37] has been shown to improve SLAM performance further. Assuming that online fine-tuning can obtain reasonable depth prediction (without extreme accuracy), we use depth prediction as a hint to cull the potential noisy map points from being included in global photometric BA, resulting in an improved SLAM accuracy. As for SLAM, we opt for SVO [10] for being a computationally lightweight on an embedded computer, compared to the other state-of-the-art sparse SLAM algorithms [8], [9].

III. Method

In this section, we outline our proposed online adaptation framework (see Figure 1). The framework consists of SLAM (Section III-A), online CNN depth adaptation (Section III-B), and global BA with adapted CNN depth (Section III-C).

A. SLAM

We use SVO 2.0 [10] (its monocular variant with edgelets) to generate a set of keyframes from a monocular image stream captured by the camera on a mobile robot. Associated with each keyframe are an image, a camera pose, and a set of sparse map points. While the keyframes are being generated, we publish two serialized channel over a local network containing the newly created keyframes (image, pose and map points) and keyframe graph (latest optimized camera poses and map points of all keyframes). The reason for publishing the keyframe graph is that BA, in most cases, improves the camera pose and map point estimations. Therefore, it is beneficial for improving the training loss in the online adaptation.

B. Online CNN depth adaptation

Our online adaptation is inspired by early stopping [38], whose goal is to avoid overfitting by stopping the fine-tuning when depth prediction is reasonably accurate, evaluated in terms of validation error (see Algorithm 1 for detailed steps). To this end, we validate the depth prediction accuracy once every m incoming keyframes to determine if the fine-tuning has converged. After determining the stopping criterion, we describe the training details of the online adaptation algorithm.

https://github.com/HeYijia/svo_edgelet

The keyframes are converted to ROS messages before they are published. See http://wiki.ros.org/msg for more information.

https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/

EarlyStopping
To adapt a depth prediction CNN on sequential data without forgetting, we improve upon the experience replay method (i.e., incorporating recent data and randomly selecting old data into the training batches \([3, 5]\)) by adding regularization. Training regularization requires the consideration of the learning of two separate tasks \(T_1\) (learn from old data) and \(T_2\) (learn from recent data). The goal is to maximize the training accuracy on both \(T_1\) and \(T_2\), ensuring that we preserve the depth prediction accuracy in the new environment. To reinforce the previously learned knowledge, we modify EWC \([30]\) regularization in the following ways. First, instead of multi-task learning consolidation, we consolidate the importance of the parameters at each training batch and use it in the next batch, which considers the adaptation progress thus far to be absorbed into the posterior probability, i.e., the importance, of the parameters. Second, to estimate the posterior probability, we compute the gradient of the negative log-likelihood of the training loss, assuming constant observation noise for the

\[
\mathcal{L}_{\text{EWC}} = \mathcal{L}_{\text{train}} + \sum_i \frac{\beta}{2} \hat{F}_i (\theta_i - \hat{\theta}_i)^2
\]

(1)

with

\[
\hat{F}_i = \max \left( \frac{\mathbf{F}}{\mathbf{F}}, \max \mathbf{F} \right),
\]

(2)

where \(j\) is the number of trained batches and \(\mathbf{F}\) the consolidated empirical Fisher information matrix at the end of every training batch. \(\hat{F}_i\) is importance of the \(i\)-th parameter, which is obtained from the averaged parameter importance

\footnote{Realistically, observation noise can occur in the violation of brightness constancy assumption and motion blur. This issue can be mitigated by incorporating the heteroscedastic aleatoric uncertainty in training (studied in \([39, 40]\) and modifying a pre-trained depth prediction CNN to predict an additional uncertainty output.}

\footnote{We explain the details in Appendix A}
matrix $F$ to ensure the preservation of the previously learned knowledge. The empirical Fisher information matrix converges to the Fisher information matrix as more training samples are incorporated \cite{11}. max$_F$ determines the maximum value of $F$.

**Algorithm 2** Online fine-tuning with parameter importance regularization

1. Init $\theta$ from a pre-trained depth prediction CNN
2. $F = 0$ \hspace{1 cm} $\triangleright$ parameter importance matrix
3. while not done do
4. \hspace{1 cm} $\theta_{cpy} = \theta$ \hspace{1 cm} $\triangleright$ Make a copy
5. \hspace{1 cm} Get latest keyframe $K_i$ from SVO
6. \hspace{1 cm} Train the CNN on $K_i$ and $K_{j \sim \{0, \ldots, (i-1)\}}$ with $L_{EWC}$ (see Eq. 1) according to $\theta$ and $F$
7. \hspace{1 cm} Consolidate the parameter importance matrix $F$ according to $\theta_{cpy}$ and $\theta$ (see Eq. 2)
8. end while

To compute the training loss ($L_{\text{train}}$), we use a semi-supervised training loss \cite{20,42}, which consists of photometric re-projection ($L_{\text{photo}}$), sparse depth ($L_{\text{sparse depth}}$) and smoothness loss ($L_{\text{smooth}}$) terms, to be defined below.

First, the photometric reprojection error $L_{\text{photo}}$ is given by:

$$L_{\text{photo}} = \frac{1}{|\Omega_s|} \sum_{p \in \Omega_s} \min_{\tilde{I}_i} \rho(I_i, I_{s \rightarrow i}, p),$$ \hspace{1 cm} (3)

$$\rho(I_i, I_{s \rightarrow i}, p) = \frac{\alpha}{2} \left( 1 - \text{SSIM}(I_i, I_{s \rightarrow i})(p) \right) + (1 - \alpha) \| I_i(p) - I_{s \rightarrow i}(p) \|_1,$$ \hspace{1 cm} (4)

where $\rho(\cdot, \cdot, \cdot)$ is the minimum per-pixel photometric reprojection error \cite{42} between the reconstructed images from the adjacent keyframes $s \in \{i-1, i+1\}$ and the $i$-th keyframe.

To reconstruct the target image from an adjacent image $I_{s \rightarrow i}$, we use bilinear sampling at the valid ($\Omega_s$) re-projected pixel locations $p'$ using depth prediction $D_{i,\text{CNN}}$, a known camera intrinsics $K$ and relative keyframe pose transformation $T_{i \rightarrow s}$ from SVO:

$$I_{s \rightarrow i}(p) = I_s(p') \quad \forall p \in \Omega_s,$$

$$p' \sim KT_{i \rightarrow s}D_{i,\text{CNN}}(p)K^{-1}p.$$ \hspace{1 cm} (5)

Then, to define $L_{\text{sparse depth}}$, we project the sparse map points generated by SVO in each keyframe to form sparse depth maps as ground truth labels \cite{20,43}:

$$L_{\text{sparse depth}} = \frac{1}{|\Omega_{i,\text{sparse}}|} \sum_{p \in \Omega_{i,\text{sparse}}} \left| \frac{1}{D_{i,\text{CNN}}(p)} - \frac{1}{D_{i,\text{sparse}}(p)} \right|.$$ \hspace{1 cm} (6)

where $D_{i,\text{sparse}}$ is the sparse depth map of keyframe $i$, $\Omega_{i,\text{sparse}}$ a set of re-projected (sub)pixel locations containing depth values of their corresponding map points from SVO, and $|\Omega_{i,\text{sparse}}|$ the number of valid re-projections. By using inverse depth, near depth is penalized more heavily than far depth.

Finally, we compute the edge-aware smoothness loss $L_{\text{smooth}}$ using the image gradient:

$$L_{\text{smooth}} = \frac{1}{N} \sum_p |\partial_x D_{i,\text{CNN}}(p)| e^{-|\partial_x I_i(p)|} + |\partial_y D_{i,\text{CNN}}(p)| e^{-|\partial_y I_i(p)|}$$ \hspace{1 cm} (7)

where $\partial$ is the gradient operator.

Combining the three loss terms, the final training loss is

$$L_{\text{train}} = L_{\text{photo}} + \lambda_1 L_{\text{sparse depth}} + \lambda_2 L_{\text{smooth}}$$ \hspace{1 cm} (8)

and averaged across the most recent keyframe and a randomly selected old keyframe (experience replay). $\lambda_1$ and $\lambda_2$ are the weighting parameters. Algorithm 2 details the steps involved in the online fine-tuning. To validate the training accuracy, we use the sparse depth loss term:

$$\mathcal{L}_{\text{val}} = \mathcal{L}_{\text{sparse depth}}.$$ \hspace{1 cm} (9)

C. Global BA

To optimize the structure and motion, we employ the traditional photometric BA \cite{37}, which jointly optimize the keyframe poses and map points. However, we found that the map points generated by SVO, albeit the sparsest amongst the state-of-the-art visual SLAM algorithms \cite{8,9}, still contain noisy and redundant map points. Therefore, with the learned depth information, we introduce a map point culling step before performing global photometric BA. To determine if a map point should be culled, we identify a host keyframe for the map point (see Figure 2) and check if the depth of the map point is within the correctness range:

$$g(d_{mp}, d_{CNN}) = \begin{cases} 1, & \text{if } |d_{mp} - d_{CNN}| < \gamma d_{CNN} \text{ or } d_{CNN} > d_{\text{max}} \\ 0, & \text{otherwise} \end{cases}$$ \hspace{1 cm} (10)

where $d_{CNN}$ and $d_{mp}$ are the predicted CNN depth and the depth of the map point in the host keyframe, $d_{\text{max}}$ a threshold that defines an effective depth range of the CNN depth values (similar to a depth sensor) to avoid far map points from being culled prematurely. $g(\cdot)$ evaluates if the map point is valid.

Then, we use the standard photometric BA formulation \cite{37} to globally optimize the structure and camera poses (collectively defined as the SLAM variables $X$) given by:

$$X^* = \arg \min_X \frac{1}{2} \sum_k \| e_k \|^2$$ \hspace{1 cm} (11)

with

$$e_k = z_k - h_k(X_k)$$ \hspace{1 cm} (12)

where $e_k$ is the photometric error induced by a subset of the SLAM variables $X_k \subseteq X$, $z_k$ the reference image patch and $h_k(\cdot)$ the measurement model to obtain the re-projected image patch.

IV. EVALUATION

To evaluate the performance of our proposed method for online CNN adaptation experimentally, our mobile robot hardware setup consists of the following main components: a TurtleBot, an Nvidia Jetson AGX Xavier, an Orbbec Astra RGBD camera, and a laptop as shown in Figure 3. With the Jetson and laptop connected to the same wireless network, we run the SLAM process (tracking, mapping, and BA) on the Jetson and the fine-tuning process on the laptop.

\[ Specifications: \text{Intel 7820HK CPU and Nvidia GTX 1070 GPU} \]
To illustrate the effectiveness of the online adaptation, we use Monodepth2's mono+stereo_640x192 pre-trained CNN model, which has been trained on outdoor scenes fine-tuned in an indoor environment. For the network to learn, we use Adam optimizer with a learning rate of $10^{-3}$, and set the weighting of the loss function $\lambda_1, \lambda_2, \alpha$ and $\beta$ at $0.1, 0.1, 0.85$ and $5 \times 10^7$, respectively. For EWC consolidation of parameter importance matrix, $\max_{\mathcal{F}}$ is set to 0.001.

In SVO, the main change we made is the increased number of tracked features in the keyframes. We have modified the following parameter values: $\text{max}_f \text{fts} = 500$, $\text{grid} \_\text{size} = 20$, and $\text{core} \_\text{kfs} = 5$. For performing map point culling, we define the correctness range by setting $\gamma = 0.5$ and $d_{\text{max}} = 1.5$. This particular design allows for more map points to be generated for fine-tuning the depth prediction CNN. The online adapted depth is then used for removing the potential noisy map points. After the noisy point removal, each map point has a maximum of five photometric re-projections to the nearby keyframes, including the existing 3D-2D constraints (i.e., the list of observed keyframes for the map point), for performing global photometric BA in a separate thread. For each re-projection edge, we get a $3 \times 3$ image patch around the re-projected image coordinates to compute the photometric errors.

In the following, we present experimental results to validate the performance of our proposed method. Section IV-A details the dataset that we collect in our laboratory and its purposes. Section IV-B compares and contrasts our proposed online adaptation method with regularization against the state-of-the-art methods. Section IV-C conducts an ablation study to compare different online adaptation schemes in overcoming catastrophic forgetting. Section IV-D presents quantitative and qualitative results of our proposed map point culling with online adapted depth and global photometric BA to improve SLAM accuracy. Section IV-E analyzes the accuracy of online adapted and pre-trained depth prediction to show the advantage of using online adaptation to improve SLAM performance. And lastly, Section IV-F evaluates the runtimes of our proposed online adaptation framework.

A. Laboratory dataset

The reason for collecting our own dataset is that the image sequences in existing benchmarking datasets are not long enough to evaluate and illustrate the effectiveness of our proposed online adaptation. Our dataset contains two sequences (dubbed Lab1 and Lab2) that we collected in our laboratory (see Figure 5 for the images of the captured environment). We record both sequences in the same environment, with the difference being the Lab2 sequence (16366 images) is longer than the Lab1 sequence (10611 images).

B. Online adaptation

To evaluate the effectiveness of our proposed online adaptation, we compare our method against the state-of-the-art methods: a SLAM-based approach by Luo et al. [28] (Section IV-B1) and a learning-based approach by Kuznietsov et al. [4] (Section IV-B2). For the comparison, we adopt the datasets and performance metric in [28]. The datasets are the ICL-NUIM [47] and TUM RGB-D [44] datasets and the performance metric is the percentage of overall depth pixels below 10% relative errors, i.e., $\frac{|D-\text{gt}|}{\text{gt}} < 0.1$.

1) SLAM-based online adaptation: Table I compares the overall depth accuracy between our method and a similar method by Luo et al. [28]. Our method improves the overall depth accuracy by around 23% (the last two columns), compared to 4% (2nd and 3rd column) by Luo et al. [28]. The performance gap could be due to the number of keyframes used in the fine-tuning; we use all the keyframes generated by SVO [10], whereas Luo et al. [28] use the keyframe pairs by LSD-SLAM [48] that have a dominant horizontal
TABLE I: A comparison between the overall depth accuracy of our method and Luo et al.'s [28] SLAM-based online adaptation on the ICL-NUIM [47] and TUM RGB-D [44] datasets. (TUM/seq1: fr3_long_office_household, TUM/seq2: fr3_nostructure_near_withloop, TUM/seq3: fr3_structure_texture_far).

| Sequence                  | Percentage of correct depth (%) | Pre-trained [28] | Pre-trained* [28] | Ours* |
|---------------------------|---------------------------------|------------------|-------------------|-------|
| ICL/office0               | 19.117                          | 22.206           | 8.766             | 12.541|
| ICL/office1               | 28.086                          | 31.289           | 19.739            | 21.870|
| ICL/office2               | 21.695                          | 21.695           | 4.591             | 42.244|
| ICL/living0               | 18.680                          | 23.278           | 7.726             | 41.375|
| ICL/living1               | 21.071                          | 22.774           | 8.518             | 49.075|
| ICL/living2               | 16.150                          | 20.995           | 13.509            | 25.159|
| TUM/seq1                  | 18.208                          | 20.259           | 10.999            | 23.861|
| TUM/seq2                  | 25.796                          | 29.014           | 12.678            | 52.162|
| TUM/seq3                  | 20.668                          | 30.156           | 10.295            | 37.848|
| Average                   | 21.052                          | 24.630           | 10.758            | 34.015|

* Scaled to ground truth depth

motion as simulated static stereo pairs. Besides, Luo et al. [28] perform online adaptation using most recent keyframes only, which performs the worst in the online adaptation scheme comparison (see Section IV-C). Note that Luo et al. [28] evaluate the keyframe depth accuracy based on the fine-tuned CNN models in different fine-tuning stages, which may result in different overall depth correctness should the final fine-tuned CNN model be used in the evaluation.

2) Learning-based online adaptation: Table II compares our method against a learning-based online adaptation method, CoMoDa [4]. Overall, our proposed SLAM-based method outperforms CoMoDa by around 5% (the last two columns). One particular challenge in end-to-end online adaptation is the simultaneous fine-tuning of depth and pose prediction CNNs, and the pose prediction CNN is not trained in the tested indoor environments; on the contrary, we obtain camera poses from SVO [10]. Furthermore, accurate camera pose estimation is essential when performing online adaptation (see Equation 3). Plus, CoMoDa does not use regularization in online adaptation, which is a reason that it performs worse than our method (see Section IV-C).

C. Learning against catastrophic forgetting in online adaptation

To evaluate the effectiveness of our proposed regularization in online adaptation, we perform an ablation study to measure the impact of different adaptation schemes on alleviating catastrophic forgetting. To this end, we examine the overall depth accuracy (using the same 10% relative error as the metric) on the ICL-NUIM [47] and TUM RGB-D [44] datasets.

Table II shows that performing online adaptation using only most recent keyframes has the worst overall depth accuracy (3rd column), compared to using the most recent keyframe with experience replay and regularization (the last two columns). Moreover, Figure 5 illustrates catastrophic forgetting on the lr_ktl1 sequence, where the CNN is overfitted to the most recent keyframes in the sequence. In general, a combination of experience replay and regularization in online adaptation has the best overall depth accuracy (see the last column in Table II and Figure 4). Thus, it validates the effectiveness of our proposed online adaptation scheme. However, for a short sequence, regularization can inhibit changes in the CNN parameters and hurt the online adaptation accuracy (2nd last row in Table II). For the sake of completeness, we compare the online adaptation performances using the three popular regularization techniques: EWC [30], MAS [31] and SI [32], and they perform similarly.

D. Effect of global photometric BA with map point culling on SLAM accuracy

After the depth prediction CNN has been fine-tuned to a reasonable accuracy, our following proposed method incorporates the online adapted depth into removing potential noisy map points and then performs photometric global BA to improve SLAM performance.

1) Camera tracking error: To measure the camera tracking performance, we use the standard absolute trajectory RMSE

2) SLAM performance:

Equation 5). Plus, CoMoDa does not use regularization in online adaptation (see Section IV-C). Thus, it validates the effectiveness of our proposed online adaptation scheme. However, for a short sequence, regularization can inhibit changes in the CNN parameters and hurt the online adaptation accuracy (2nd last row in Table II). For the sake of completeness, we compare the online adaptation performances using the three popular regularization techniques: EWC [30], MAS [31] and SI [32], and they perform similarly.

D. Effect of global photometric BA with map point culling on SLAM accuracy

After the depth prediction CNN has been fine-tuned to a reasonable accuracy, our following proposed method incorporates the online adapted depth into removing potential noisy map points and then performs photometric global BA to improve SLAM performance.

1) Camera tracking error: To measure the camera tracking performance, we use the standard absolute trajectory RMSE

2) SLAM performance:

Fig. 4: A comparison of different online adaptation schemes tested on the ICL-NUIM [47] of_kt3 (left) and lr_ktl1 (right) using the final online adapted CNN. Adaptation accuracy is measured by the averaged percentage of overall depth accuracy over all frames up to the frame. Method 1: fine-tuning on most recent keyframes only; Method 2: fine-tuning on the most recent keyframe with experience replay; Method 3: fine-tuning on the most recent keyframe with regularization; Method 4: fine-tuning on the most recent keyframe with experience replay and regularization.
TABLE III: An ablation study on different online adaptation schemes using our proposed framework on the ICL-NUIM [47] and TUM RGB-D [44] datasets. (TUM/seq1: fr3_long_office_household, TUM/seq2: fr3_nostructure_texture_near_withloop, TUM/seq3: fr3_structure_texture_far)

| Sequence        | Pre-trained | Ours(a) | Ours(b) | Ours(c) |
|-----------------|-------------|---------|---------|---------|
| ICL/office0     | 8.766       | 7.249   | 12.734  | 12.541  |
| ICL/office1     | 19.739      | 4.034   | 22.613  | 21.870  |
| ICL/office2     | 4.591       | 21.449  | 34.609  | 42.244  |
| ICL/living0     | 7.726       | 13.179  | 22.883  | 41.375  |
| ICL/living1     | 8.518       | 38.246  | 41.149  | 49.075  |
| ICL/living2     | 13.509      | 9.110   | 37.539  | 52.162  |
| TUM/seq1        | 10.999      | 16.012  | 23.861  | 25.159  |
| TUM/seq2        | 12.678      | 14.457  | 37.539  | 52.162  |
| TUM/seq3        | 10.295      | 49.515  | 51.545  | 37.848  |
| Average         | 10.758      | 19.250  | 28.441  | 34.015  |

(a) Online adaptation with two most recent keyframes
(b) Online adaptation with the most recent keyframe and experience replay
(c) Online adaptation with the most recent keyframe, experience replay and regularization

TABLE IV: A comparison of the camera tracking ATEs with and without global photometric BA on the ICL-NUIM [47] and TUM RGB-D [44] datasets. (TUM/seq1: fr3_long_office_household, TUM/seq2: fr3_nostructure_texture_near_withloop, TUM/seq3: fr3_structure_texture_far)

| Sequence        | Without BA* | With BA* |
|-----------------|-------------|----------|
| ICL/office0     | 0.363       | 0.323    |
| ICL/office1     | 0.196       | 0.204    |
| ICL/office2     | 0.170       | 0.171    |
| ICL/living0     | 0.220       | 0.227    |
| ICL/living1     | 0.032       | 0.026    |
| ICL/living2     | 0.161       | 0.158    |
| TUM/seq1        | 0.048       | 0.073    |
| TUM/seq2        | 0.029       | 0.012    |
| TUM/seq3        | 0.011       | 0.012    |
| Average         | 0.141       | 0.134    |

*Aligned to ground truth trajectory

(ATE) as the performance metric on the ICL-NUIM [47] and TUM RGB-D [44] datasets, which contains ground truth camera poses for the evaluation. Table IV compares the ATEs obtained from SVO and SVO with our proposed map point culling and global photometric BA. In general, performing global photometric BA leads to a lower overall ATE than that of without the global BA (compare the last two columns in Table IV). Our proposed photometric global BA with map point culling reduces the ATE by more than 50% (from 0.029 to 0.011) on the TUM/seq2 sequence, likely due to the better online adaptation of depth prediction (check the corresponding rows in Table III).

2) Map reconstruction error: To highlight the effectiveness of the online adaptation, we evaluate the accuracy of the SVO map on long sequences, Lab1 and Lab2, from our Laboratory dataset. We use the scale-invariant inverse depth error as the performance metric [49] given by:

\[ e_{si} = \sqrt{\frac{1}{N} \sum_i d_i^2 - \frac{1}{N^2} \left( \sum_i d_i \right)^2 } \]  \hspace{1cm} (13)

where \( d_i = \log z_i - \log z_i^* \), and the superscript * indicates the ground truth depth.

Fig. 5: Qualitative comparisons between different correctness thresholds used in map point (MP) culling: (a) no culling, (b) MP culling with \( \alpha = 0.5 \), (c) MP culling with \( \alpha = 0.25 \) and (d) MP culling with \( \alpha = 0.15 \).

To allow for a more accurate online adaptation accuracy, we increase the number of map points by tweaking the \( \text{max}_\text{fts} \) parameter in SVO. Table V shows that the SVO map accuracy is improved by our proposed map point culling with online adapted depth prediction and global photometric BA (compare the \( e_\text{map} \) with and without BA in Table IV and Figure 5(a) and 5(b)). The improved map accuracy also comes at the expense of removing good map points, which can be mitigated by having a training scheme with better online adaptation performance. As a result, our proposed online adaptation framework achieves a mutual adaptation of depth prediction and visual SLAM.

TABLE V: A comparison of the number of points and depth error of the SVO map with and without our proposed map point culling and global photometric BA. A larger \( \text{max}_\text{fts} \) generates more map points in SVO.

| BA culling | SVO param | Lab1 | Lab2 |
|------------|-----------|------|------|
| ✓          | ✓         | 120  | 14469|
| ✓          | ✓         | 7435 | 20464|
| ✓          | ✓         | 19179| 79556|
| ✓          | ✓         | 500  | 9226 |
| ✓          | ✓         | 500  | 23743|
| ✓          | ✓         | 9916 | 0.493|

Note: \( \text{max}_\text{fts} \) is set at 120 by default in SVO.
Table VI: A comparison of depth errors of MiDaS [26], DiverseDepth [27] and our online adapted Monodepth2 [42] CNN model.

| Method            | Lab1   | Lab2   |
|-------------------|--------|--------|
| MiDaS (V2.1)      | 0.271  | 0.256  |
| MiDaS (V2.3.0)    | 0.315  | 0.322  |
| DiverseDepth      | 0.450  | 0.435  |
| Pre-trained       | 0.401  | 0.399  |
| Ours              | 0.235  | 0.219  |

† only the re-projected (sub)pixels

E. Online adaptation vs. relative depth prediction

A competing idea to our online adaption for resolving the domain gap in single-image depth prediction is to train a network with a large number of datasets across multiple domains to improve its generalization. Unlike absolute depth prediction, which is trained in a narrow domain, state-of-the-art relative depth prediction CNN models, such as MiDaS [26] and DiverseDepth [27], have been trained on an extensive collection of datasets. Therefore, to compare between online adapted depth and pre-trained depth prediction, we use the same scale-invariant inverse depth error \( e_{si} \) described in Section IV-D2.

Table VI shows a quantitative comparison of the depth errors of MiDaS, DiverseDepth and our fine-tuned Monodepth2 CNN model. On both sequences (2nd and 3rd column), our method has the lowest depth error in comparison with the relative depth prediction CNNs (MiDaS and DiverseDepth) (see Figure 6 for a qualitative comparison). Due to the similarity between Lab1 and Lab2, the depth prediction errors of MiDaS, DiverseDepth and pre-trained Monodepth2 are similar in both sequences. Conversely, our method can further boost the depth prediction accuracy (see the last row of Table VI) resulting from our proposed online adaptation, especially on the longer sequence (Lab2).

F. Runtime evaluation

The two main time-consuming processes in our proposed online adaptation framework are depth prediction and fine-tuning. On average, depth prediction and one fine-tuning iteration take about 18 ms and 250 ms to process, respectively. Incidentally, the global photometric BA time consumption varies according to the total number of photometric re-projections between the map points and keyframes, but real-time processing should not be affected as the BA process itself runs on another thread.

V. Conclusion

In this paper, we have addressed a practical robotics problem concerning the use of single-image depth prediction to improve SLAM performance. Particularly, we have proposed a novel online adaptation framework in which the fine-tuning is enhanced with regularization for retaining the previously learned knowledge while the CNN is being trained continually.

Furthermore, we have investigated the use of regularization in the learning of depth prediction, a single-task regression problem, with quantitative results and an ablation study to show its effectiveness against catastrophic forgetting. Also, we have demonstrated the use of fine-tuned depth prediction for map point culling before running global photometric BA, resulting in improved camera tracking and map reconstruction. Lastly, we have compared our online adaptation framework against state-of-the-art depth prediction CNNs that have been trained on a large number of datasets across domains, showing that our online adapted depth prediction CNN has a lower depth error, especially after performing online adaptation on a long sequence.

APPENDIX

A. Adapted EWC regularization for single-task regression problem

To perform online adaptation incrementally on a single task, we measure and consolidate the importance of the CNN parameters at each iteration, allowing for the preservation of learned knowledge. The EWC loss comprises the training loss and a regularizer to fine-tune the CNN parameters while preserving the important parameters measured through consolidation.

Formally, let \( E \) be the loss function of the problem at hand and assuming \( E \) to be locally smooth, we may expand the function about \( \theta_0 \) using Taylor series approximation:

\[
E(\theta) = E(\theta_0) + J_\theta(\theta_0)(\theta - \theta_0) + \frac{1}{2!}(\theta - \theta_0)^T H_\theta(\theta_0)(\theta - \theta_0) + \text{h.o.t.} \tag{14}
\]

where \( J_\theta(\theta_0) \) and \( H_\theta(\theta_0) \) are the Jacobian and Hessian of \( E \) evaluated at \( \theta_0 \), respectively. Assume that the CNN is fine-tuned on old data \( T_1 \), which implies that \( \theta_0 \) would be at a local minimum (denoted by \( \theta^*_0 \)) and that \( E'(\theta^*_0) = 0 \), we can rewrite Equation (14) as

\[
E_{T_1}(\theta) \approx E(\theta^*_0) + \frac{1}{2!}(\theta - \theta^*_0)^T H_\theta(\theta^*_0)(\theta - \theta^*_0) \tag{15}
\]

where \( E_{T_1} \) is the approximated error for old data \( T_1 \). Next, to fine-tune the CNN on recent data \( T_2 \), we can use the following loss function:

\[
E(\theta) = E_{T_2}(\theta) + \beta E_{T_1}(\theta), \tag{16}
\]

where \( \beta \) controls the weighting between the two loss terms.

Substituting Equation (15) into Equation (16) we get

\[
E(\theta) = E_{T_2}(\theta) + \beta \left( E(\theta^*_0) + \frac{1}{2!}(\theta - \theta^*_0)^T H_\theta(\theta^*_0)(\theta - \theta^*_0) \right)
= E_{T_2}(\theta) + \beta \left( (\theta - \theta^*_0)^T H_\theta(\theta^*_0)(\theta - \theta^*_0) \right), \tag{17}
\]

\footnote{For relative depth prediction to achieve maximum accuracy, accurate scale- and shift-correction for each predicted depth map is required [27, 50].}
Fig. 6: A qualitative comparison of the back-projected point clouds (shown in black) between (from left to right) ground truth depth with SVO map points (in blue), MiDaS v2.1, MiDaS v3.0, DiverseDepth, pre-trained Monodepth2, and our online adapted Monodepth2. From top to bottom: first and second viewpoint of the back-projected depth maps and the predicted depth maps by the aforementioned CNN models. The predicted depth maps are scaled to ground truth. Best viewed digitally.
in which the first term of Equation 15 is being treated as a constant and can be eliminated in the optimization. To measure the importance of the parameters contained in the Hessian matrix, Kirkpatrick et al. approximate the Gaussian distribution of the posterior with Laplace approximation, whereby diagonal of the Fisher information matrix replaces the Hessian matrix for the approximation of the posterior 30:

$$E(\theta) = E_{\tau}(\theta) + \frac{\beta}{2} \left( \sum_i \Omega_i(\theta_i - \theta_{\Omega,i})^2 \right), \quad (18)$$

**B. SI and MAS regularizations**

Similar to EWC regularization, SI and MAS can be expressed using the following loss function:

$$L = L_{\text{train}} + \lambda \sum_i \Omega_i(\theta_i - \theta_{\Omega,i})^2, \quad (19)$$

with the main difference being the estimation of the weight importance matrix, $\Omega_i$. For MAS, the weight importance matrix is given by the gradient of the squared $L_2$-norm of the CNN output function, $G$ 31:

$$\Omega_i = \frac{1}{N} \sum_{j=1}^{N} \frac{\partial ||G(x_j, \theta)||_2^2}{\partial \theta_i}, \quad (20)$$

whereas for SI, the weight importance matrix is given by 32:

$$\Omega_i = \sum_j \frac{\omega_{ij}}{(\Delta \theta_{ij})^2 + \xi}, \quad (21)$$

where $\omega_{ij}$ is the estimated per-parameter contribution to the total loss, and $\Delta \theta_{ij}$ the total trajectory of the parameter.

**REFERENCES**

[1] Matthais Delath, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[2] German I Parisi, Ronald Kemker, Jose L Par, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113, 2019.

[3] Mila Knowles, Valentin Peretroukhin, W Nicholas Greene, and Nicholas Roy. Toward robust and efficient online adaptation for deep stereo depth estimation. In *International Conference on Robotics and Automation (ICRA)*, pages 12292–12198, 2021.

[4] Yevhen Kuznietsov, Marc Proesmans, and Luc Van Gool. Comoda: Continuous monocular depth adaptation using past experiences. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2907–2917, January 2021.

[5] Zhenyu Zhang, Stéphane Lathuilière, Elisa Ricci, Nicu Sebe, Yan Yan, and Jian Yang. Online depth learning against forgetting in monocular and pattern recognition videos. In *International Conference on Robotics and Automation (ICRA)*, pages 4009–4018, 2021.

[6] Visor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Ales Leonardis. Vitamin-e: Visual tracking and mapping with extremely dense feature points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2019.

[7] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE Transactions on Robotics*, 2021.

[8] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.
Shing Yan Loo, Syamsiah Mashohor, Sai Hong Tang, and Hong Zhang. Deeprelativefusion: Dense monocular slam using single-image relative depth prediction. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021.

Moein Shakeri received the M.S. degree in computer engineering from Ferdowsi University, Iran in 2009, and Ph.D. degree in computing science from University of Alberta, Canada, 2019. During his Ph.D. he has been awarded several grants, and Ph.D outstanding thesis award. He is currently working as a post-doctoral research fellow in the department of computing science at the University of Alberta. His research interests include dense map reconstruction, loop closure detection, object detection and segmentation, and polarimetric 3D reconstruction.

Professor Hong Zhang received his B.S. from Northeastern University (Boston) in 1982, and Ph.D. from Purdue University in 1986, both in Electrical Engineering. Subsequently he conducted post-doctoral research at the University of Pennsylvania before he joined the Department of Computing Science at the University of Alberta, Canada, where he worked for over 30 years. Since October 2020, he has been a Chair Professor in the Department of Electronic and Electrical Engineering at the Southern University of Science and Technology in Shenzhen, China.