# Deep Probabilistic Feature-Metric Tracking

Binbin Xu, *Graduate Student Member, IEEE*, Andrew J. Davison, and Stefan Leutenegger

***Abstract*—Dense image alignment from RGB-D images remains a critical issue for real-world applications, especially under challenging lighting conditions and in a wide baseline setting. In this letter, we propose a new framework to learn a pixel-wise deep feature map and a deep feature-metric uncertainty map predicted by a Convolutional Neural Network (CNN), which together formulate a deep probabilistic feature-metric residual of the two-view constraint that can be minimised using Gauss-Newton in a coarse-to-fine optimisation framework. Furthermore, our network predicts a deep initial pose for faster and more reliable convergence. The optimisation steps are differentiable and unrolled to train in an end-to-end fashion. Due to its probabilistic essence, our approach can easily couple with other residuals, where we show a combination with ICP. Experimental results demonstrate state-of-the-art performances on the TUM RGB-D dataset and the 3D rigid object tracking dataset. We further demonstrate our method's robustness and convergence qualitatively.**

***Index Terms*—Deep learning for visual perception, SLAM.**

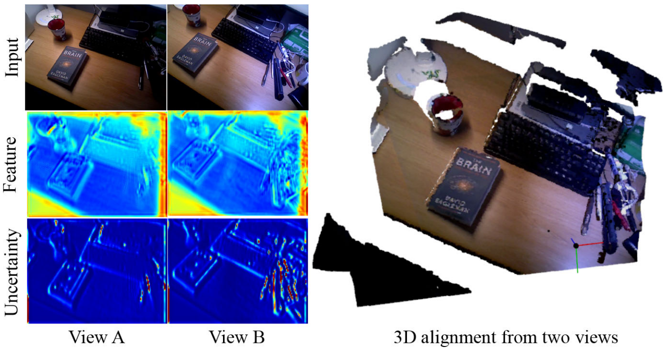

Fig. 1. We propose a probabilistic feature-metric tracking method that estimates dense feature and uncertainty maps from a pair of RGB-D images to optimise the relative pose between them. Our method can handle strong lighting changes and large motion scenarios by leveraging features that are robust to lighting changes, e.g. on the desk surface, and predicting high uncertainties on areas that the network cannot handle, e.g. for the strong lighting changes near the pens.

## I. Introduction

Dense image alignment [1] using the photometric residual has been widely applied in 2D tracking [2], 3D object tracking [3], optical flow [4], and SLAM [5]. In visual SLAM, it leads to two types of estimator designs: sparse [6] and dense type [5]. There has been an argument that dense methods that utilise information from all image pixels should exhibit better performance in terms of robustness and accuracy. However, this is not necessarily the case in reality, as investigated in [7], especially compared to performance achieved by systems using the indirect sparse residual formulation (reprojection error) [8].

One reason is that lighting change and reflection in real scenes break the brightness constancy assumption [4] commonly used in dense image alignment. Thus the resulting dense photometric residual cannot be well explained by the Gaussian distribution assumed in the Gauss-Newton scheme, which is in contrast to reprojection error minimisation that may still work robustly as long as sparse feature matches can be established. Secondly, the photometric residual considers only very local color consistency, which requires a good initialisation close to the global minimum. This leads to a poorer estimation accuracy when the baseline gets larger. On the contrary, the keypoint reprojection residual models a global constraint using a sparse feature descriptor, leading to better convergence properties.

In this letter, we are trying to address these issues by replacing raw intensity image alignment with deep feature map alignment. Different from the existing learning-based feature-metric alignment [9]–[13], we argue that the feature-metric residual should incorporate not simply the feature difference but also the corresponding uncertainty. Predictions from neural networks inherently are uncertain, which can be estimated [14]. Secondly, and also importantly, SLAM has most successfully been posed as a probabilistic problem, where uncertainty of the residuals has to be known [15], in particular when fusing different sensors and residuals. We will show how our feature-metric residuals can be combined with geometric ICP residuals using uncertainties to further improve results. The proposed probabilistic feature-metric residuals are minimised using coarse-to-fine Gauss-Newton optimisation. To ensure that the learned feature-metric cost landscape is suitable for the Gauss-Newton optimisation, we unroll the iterative optimisation steps and train the whole pipeline end-to-end. To handle the initialisation issue in the wide baseline case, we include training pairs with varied baselines and propose to replace the identity initialisation with a predicted initial pose from a pose network. This can improve the system convergence by bringing the initialisation into the convergence basin of the correct minimum. As shown in Fig. 1, the proposed method can handle large motion and strong illumination variance. The learned features are robust to lighting

Manuscript received September 1, 2020; accepted November 13, 2020. Date of publication November 19, 2020; date of current version December 1, 2020. This letter was recommended for publication by Associate Editor F. Fraundorfer and Editor S. Behnke upon evaluation of the Reviewers' comments. This work was supported by Imperial College London, and the EPSRC Grant Aerial ABM EP/N018494/1.The work of Binbin Xu was further supported by a China Scholarship Council-Imperial Scholarship. *(Corresponding author: Binbin Xu.)*

The authors are with the Department of Computing, Imperial College London, London SW7 2BU, United Kingdom (e-mail: b.xu17@imperial.ac.uk; ajd@doc.ic.ac.uk; s.leutenegger@imperial.ac.uk).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2020.3039216, provided by the authors.

Digital Object Identifier 10.1109/LRA.2020.3039216

changes in most regions, e.g. reflection on desk surface, and the uncertainty map (red means high uncertainty) can downweigh the region, e.g. pens, where the feature predictions are uncertain. In summary, we make the following contributions:

1) We propose a dense probabilistic feature-metric residual, where a CNN predicts both feature and uncertainty maps used for non-linear least-squares minimisation to estimate the relative camera or object pose.
2) In our CNN architecture, we propose a coupled feature encoder and pose predictor network, which combines the learning-based initial pose prediction and the learned features/uncertainties for pose optimisation, and train them together end-to-end.
3) We further demonstrate how our proposed probabilistic feature-metric residual can easily lend itself to integration with other residuals, where a classic ICP residual is showcased.

We evaluate our proposed method on the TUM RGB-D SLAM dataset [16] and MovingObjects3D rigid motion dataset [10]. We provide ablation studies to validate each contribution component. We further provide a qualitative evaluation on the convergence basin and demonstrate the robustness under strong lighting changes.

## II. Related Work

**Feature-metric Alignment:** To relax the brightness constancy constraint in direct image alignment, several recent works have exploited the feature-metric alignment by utilising features from neural networks. [9], [17] replace image intensity with high-dimensional features extracted from a pre-trained neural network for tracking and show a better robustness than using image intensity. However, the pre-trained features are not naturally consistent across different views and the redundancy in the pre-trained very high-dimensional features means a high cost of memory and computation time.

[12] proposes to learn a robust feature descriptor suitable for estimating dense correspondence in different lighting conditions and viewpoints using the contrastive loss [18]. [13] combines the contrastive loss with a Gauss-Newton loss, which includes a 2-dimensional pixel position uncertainty, to train dense features. However, both of these works generate a feature map good for correspondence matching rather than alignment. The composed residuals do not necessarily fit well with the least square optimisation used for pose estimation. This is why [12] requires a RANSAC step for refinement and [13] is only used for re-localisation.

Recently, some methods start to explore how to combine the feature map learning more tightly with the least-square optimisation of camera tracking, based on the differentiable property of iterative optimisation. [19] learn feature maps for 2D image tracking in the Lucas-Kanade framework. [11] propose feature-metric bundle adjustment for 3D reconstruction. [20] propose to use feature maps for depth prediction and pose estimation. However, these works only consider a spatial correlation in feature generation, ignoring the temporal correlation in input image pairs. Quite related to our work, [10] propose a spatio-temporal feature encoder by concatenating two views for the network input and further propose an m-estimator network and damping network for pose optimisation. However, different from ours, none of these works exploit feature-metric uncertainty in their settings, nor combine a pose predictor to boost convergence.

**Deep Pose Prediction:** A different way to estimate pose from a pair of images is to leverage CNN predictions directly [21], [22]. Learning a direct mapping from input images to 6D relative pose skips potential convergence issues of least-squares optimisation. However, it requires a large number of model parameters and a vast amount of training data, while not necessarily generalising to new scenes.

To improve accuracy and generalisation, some recent works include coarse-to-fine estimation [23] and iterative refinement [24] to estimate a relative transformation. Despite some shared weights in iterations, these works still come with a much larger model capacity (i.e. parameter number) than the ones using optimisation – even those with learned features – and do not necessarily show an advantage in terms of pose accuracy. To better leverage both types of approaches, we propose a coarse-to-fine optimisation using learned features and uncertainties, plus a direct pose prediction on the coarsest layer serving as an *initial guess*, which takes the output from the coarsest level two-view encoder as an input to make it compact.

**Uncertainty Learning:** Safety considerations have prompted recent works on uncertainty estimation of deep learning, as discussed in [14] and applied to several tasks [25]. [26] fuse the predicted depth into a monocular SLAM system and estimate the depth uncertainties via its difference with the nearest key-frame. [23] propose to estimate both depth and pose uncertainty in their depth and pose prediction networks. [27] formulate the depth uncertainty differently using a probability volume. Recently, D3VO [28] propose to estimate the photometric uncertainties and predict a relative pose to initialise the pose optimisation. Most of these works, if not all, model the uncertainty based on the difference between the prediction and the ground truth values. In contrast to these works, we propose a novel feature-metric uncertainty and learn it without ground truth feature maps available in the training. Instead, we formulate the uncertainty in a novel probabilistic feature-metric residual and learn it implicitly as part of the least-squares optimisation. The learned features and uncertainties should lead to a better optimised pose via training back-propagation.

## III. Method

Fig. 2 shows an overview of our system. For a pair of RGB-D frames, frame $A$ $\underrightarrow{\mathscr{F}}_A$ and frame $B$ $\underrightarrow{\mathscr{F}}_B$, our aim is to estimate its relative transformation $T_{AB} = (C_{AB}, AAB) \in (SO(3) \times \mathbf{R}^3)$, from $\underrightarrow{\mathscr{F}}_B$ to $\underrightarrow{\mathscr{F}}_A$. We represent $T_{AB}$ in twist coordinates $\boldsymbol{\xi}$ by $T_{AB}(\boldsymbol{\xi}) = \exp(\boldsymbol{\xi}_{AB})$. Each frame has a depth map $\mathbf{D}$ and a color image $\mathbf{I}$. The network components in our whole system are denoted as $\phi$, with the two-view spatio-temporal encoder $\phi_\theta$, the feature encoder $\phi_F$, the uncertainty encoder $\phi_\sigma$, and the pose network $\phi_T$. The weights are shared across the two views for $\phi_\theta$, $\phi_F$, and $\phi_\sigma$. The architecture details of all our network components can be found in the subsection III-E.

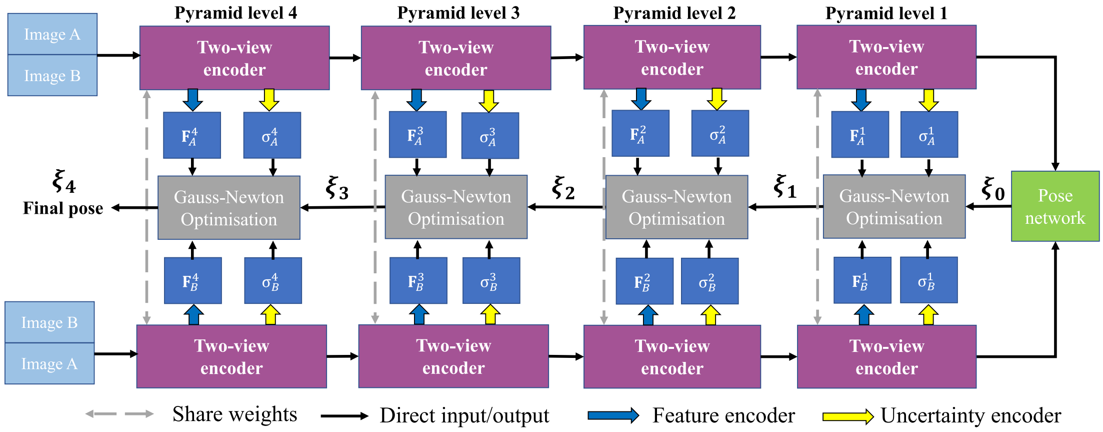

Fig. 2. Overview of our proposed deep probabilistic feature-metric tracking method. For two views, we input image $A$ and image $B$, by concatenating them as $\{A, B\}$ and $\{B, A\}$, respectively, to our two-view encoder pyramid network. At each pyramid level, we extract the output from the two-view encoder and feed it into the feature encoder and uncertainty encoder separately to extract dense feature and uncertainty maps. Then we optimise the pose by minimising the proposed probabilistic feature-metric residual, which is initialised by the pose from the coarser level. On the coarsest level, we concatenate the outputs of the two views from the two frames and run through the pose network to obtain an initial pose prediction.

To extract the spatial and temporal correlation between two frames, we first concatenate the input colour and depth image along the feature channel and feed them through the two-view spatio-temporal encoder pyramid network:

$$\mathbf{W}_A^i = \phi_\theta(\{\mathbf{I}_A, \mathbf{D}_A, \mathbf{I}_B, \mathbf{D}_B\}),\ \mathbf{W}_B^i = \phi_\theta(\{\mathbf{I}_B, \mathbf{D}_B, \mathbf{I}_A, \mathbf{D}_A\}), \tag{1}$$

where $\mathbf{W}_A^i$ and $\mathbf{W}_B^i$ are the outputs of the two-view encoder at level $i$, $i \in 1, 2, 3, 4$, for frame $A$ and $B$ respectively and $\{,\}$ is the concatenation operation. On each pyramid level, we extract the dense feature and uncertainty maps by feeding the two-view encoder outputs into the feature encoder branch and the uncertainty encoder branch:

$$\mathbf{F}_X^i = \phi_F(\mathbf{W}_X^i), \qquad \sigma_X^i = \phi_\sigma(\mathbf{W}_X^i), \tag{2}$$

where $X \in A, B$. $i$ will be omitted later when we explain operation on the same pyramid level. Different from [10] which averages the output features map into one single channel, we maintain a same high-dimensional feature map at different pyramid levels. This choice is motivated by the hypothesis that higher dimensionality should lead to higher discriminative power of the features – which we support in the experimental section.

### A. *Probabilistic Feature-Metric Residual for Pose Estimation*

In probabilistic estimation that assumes an underlying Gaussian distribution of the residuals, we equivalently minimise the weighted least squares, with the inverse covariance matrix acting as the weight. Given the dense feature and uncertainty maps on two views and an estimated pose $\boldsymbol{\xi}_{AB}$, we propose a probabilistic feature-metric residual as an uncertainty-normalised feature difference:

$$\mathbf{r}_f(\boldsymbol{\xi}_{AB}) = \frac{\bar{\mathbf{r}}_f(\boldsymbol{\xi}_{AB})}{\sigma_f(\boldsymbol{\xi}_{AB})} = \frac{\mathbf{F}_A[\boldsymbol{u}_A(\boldsymbol{\xi}_{AB})] - \mathbf{F}_B[\boldsymbol{u}_B(\boldsymbol{\xi}_0)]}{\sqrt{\sigma_A^2[\boldsymbol{u}_A(\boldsymbol{\xi}_{AB})] + \sigma_B^2[\boldsymbol{u}_B(\boldsymbol{\xi}_0)]}}, \tag{3}$$

where $\boldsymbol{u}_A$ and $\boldsymbol{u}_B$ are a pair of pixel correspondences on the two frames. $\boldsymbol{u}$ represents image pixel coordinates. $\boldsymbol{u}_B(\boldsymbol{\xi}_0)$ means $\boldsymbol{u}_B$ is perturbed under zero transformation $\boldsymbol{\xi}_0$. $\bar{\mathbf{r}}_f$ is the feature difference between the correspondences on the feature map and $\sigma_f$ is the joint uncertainty estimate for the correspondence that we obtain as a combination from the individual uncertainties. Note that this assumes isotropic uncertainty w.r.t. each feature dimension – a simplification we chose (for speed) that may be revisited. Eq. 3 encourages the feature map from two different views to be as similar as possible while downweighs the features that the network is uncertain about from the either view with the predicted uncertainties. As shown in example Fig. 1, the trained features are robust to moderate lighting, reflection and view perspective variances and the trained uncertainties handle the uncertain features caused by the extreme lighting changes (lower right corner). The dense correspondence lookup is implemented via warping from frame $B$ to frame $A$ through $\boldsymbol{\xi}_{AB}$, which can be defined as:

$$\boldsymbol{u}_A(\boldsymbol{\xi}_{AB}) = \pi(T_{AB}(\boldsymbol{\xi})\pi^{-1}(\boldsymbol{u}_B, D_B[\boldsymbol{u}_B])), \tag{4}$$

where [.] represents the pixel lookup (including bilinear interpolation). $\pi$ and $\pi^{-1}$ denote the projection function to the image plane and the back-projection function to 3D (homogeneous) coordinates, respectively. By inserting Eq. 3 into a Lucas-Kanade framework [1], we formulate the pose estimation problem of an optimal pose $\boldsymbol{\xi}^*$ as:

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \frac{1}{2} \sum_{\boldsymbol{u}_B \in \mathscr{U}} \mathbf{r}_f^T(\boldsymbol{\xi})\mathbf{r}_f(\boldsymbol{\xi}), \tag{5}$$

i.e. summing all residuals over non-occluded pixels in $B$, $\mathscr{U}$, which can be iteratively solved by e.g. the Gauss-Newton method. To speed up the computation, we choose the inverse compositional formulation [29] that updates poses by applying the incremental pose on frame $B$. It allows for a more efficient computation of the feature-metric Jacobians. In each iteration, the pose is updated by $\Delta\boldsymbol{\xi}$ as:

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k \circ \Delta\boldsymbol{\xi}^{-1}, \tag{6}$$

$$\Delta\boldsymbol{\xi} = -(\mathbf{J}_f^T\mathbf{J}_f)^{-1}(\mathbf{J}_f^T\mathbf{r}_f). \tag{7}$$

$\mathbf{J}_f$ is the Jacobian of the probabilistic feature-metric residual $\boldsymbol{r}_f$ w.r.t. the relative pose $\boldsymbol{\xi}_{AB}$:

$$\mathbf{J}_f = \frac{\partial \mathbf{r}_f}{\partial \boldsymbol{\xi}_{AB}} = -\left( \frac{\nabla \mathbf{F}_B}{\sigma_f(\boldsymbol{\xi}_{AB})} + \frac{\bar{\mathbf{r}}_f(\boldsymbol{\xi}_{AB})\sigma_B \nabla \sigma_B}{\sigma_f^3(\boldsymbol{\xi}_{AB})} \right) \frac{\partial \boldsymbol{u}_B}{\partial \boldsymbol{\xi}_0}, \tag{8}$$

where $\nabla \mathbf{F}_B$ and $\nabla \sigma_B$ are the gradients of the feature maps and uncertainty maps along the two pixel dimensions in frame $B$, respectively. Under this formulation, only the components of $\sigma_f(\boldsymbol{\xi})$ and $\bar{\mathbf{r}}_f(\boldsymbol{\xi})$ need to be re-evaluated in each iteration, which can be shared when computing the residuals in Eq. 3. All the other components in Eq. 8 can be pre-computed to speed up the computation.

### B. Probabilistic Combination With ICP Residual

As an uncertainty-driven residual, our proposed residual can be naturally combined with other residuals. For example, we can combine it with an ICP residual to add a more geometric constraint. The combined residual equation is:

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \mathbf{r}_f^T(\boldsymbol{\xi})\mathbf{r}_f(\boldsymbol{\xi}) + w_g \mathbf{r}_g^T(\boldsymbol{\xi})\boldsymbol{\Sigma}_g^{-1}\mathbf{r}_g(\boldsymbol{\xi}), \tag{9}$$

where $\boldsymbol{r}_g$ and $\boldsymbol{\Sigma}_g$ are the ICP residual and uncertainty, respectively, and $w_g$ is the weight for ICP residual. The above equation can still be iteratively solved via the Gauss-Newton method. The detailed definitions of the ICP residual and Jacobian can be found in [30]. As there are no regularisation terms in Eq. 3, our learned uncertainty is a scale-free parameter. When combining with other residuals of different magnitudes, we need to scale them properly before fine-tuning to bootstrap the training. The scale of ICP weight $w_g$ is chosen (as $w_g = 0.01$) such that the individual Chi-square errors are of similar magnitude, after which the joint ICP/feature-metric training will scale the features and feature-metric uncertainties to be best balanced with the ICP.

### C. Coarse-to-Fine Optimisation and Initialisation

The cost functions in Eq. 5 and 9 can be optimised in a coarse-to-fine way using damped Gauss-Newton optimisations, which is applied on 4 pyramid levels, with a fixed number of rolled-out iterations, i.e. 3, on each level. We added a small damping constant in Eq. 7 to prevent the matrix inversion to be ill-conditioned. Coarse-to-fine optimisation methods are sensitive to coarse-level estimation, where the incorrect estimations will be propagated to finer levels and the iterative optimisation may get stuck in a wrong local minimum, especially in a wide-baseline setting. To tackle this issue, we train a pose network to bootstrap the optimisation by predicting an initial relative pose on the coarsest level, instead of using a conventional identity pose initialisation. To make the network compact, the concatenated outputs from the coarsest-level two-view encoder on the two frames serve as the inputs to our pose prediction network:

$$\boldsymbol{\xi}_0 = \phi_T(\{\mathbf{W}_A^1, \mathbf{W}_B^1\}). \tag{10}$$

To account for the multi-modal information on the coarse level, the deep initial pose network outputs $K$ pose hypotheses, which are parameterised as 3 Euler angles and 3D translation vectors, and a respective confidence probability for each hypothesis. The final predicted pose is the weighted average of all hypotheses.

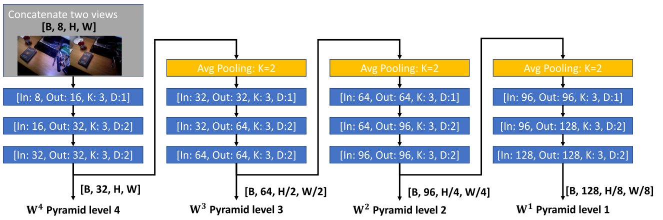
Concatenate two views
[B, 8, H, W]
[In: 8, Out: 16, K: 3, D:1]
[In: 16, Out: 32, K: 3, D:2]
[In: 32, Out: 32, K: 3, D:2]
[B, 32, H, W]
W⁴ Pyramid level 4
Avg Pooling: K=2
[In: 32, Out: 32, K: 3, D:1]
[In: 32, Out: 64, K: 3, D:2]
[In: 64, Out: 64, K: 3, D:2]
[B, 64, H/2, W/2]
W³ Pyramid level 3
Avg Pooling: K=2
[In: 64, Out: 64, K: 3, D:1]
[In: 64, Out: 96, K: 3, D:2]
[In: 96, Out: 96, K: 3, D:2]
[B, 96, H/4, W/4]
W² Pyramid level 2
Avg Pooling: K=2
[In: 96, Out: 96, K: 3, D:1]
[In: 96, Out: 128, K: 3, D:2]
[In: 128, Out: 128, K: 3, D:2]
[B, 128, H/8, W/8]
W¹ Pyramid level 1

Fig. 3. The architecture of our two-view encoder. It is composed of basic convolutional blocks (blue) and average pooling operations (yellow). The basic convolutional block is grouped by a convolutional layer and followed by a BatchNorm layer, and a ELU layer. [In, Out, K, D] represents [Input channel, Output channel, Kernel size, Dilation] with stride always being 1.

### D. Training Setup

The predicted initial pose and the estimated poses per pyramid level are compared to the ground truth pose and the resulting gradients in the optimisation are used for back-propagation to update all the learning weights. To balance influence of rotation vs. translation, we use the 3D End-Point-Error (EPE) as the training loss: given the ground truth relative transformation $T_{AB}(\boldsymbol{\xi})$ and the estimated/predicted pose $T_{AB}(\boldsymbol{\xi}_i)$, the loss is composed as:

$$L = \frac{1}{|\mathscr{V}|} \sum_{i \in \mathscr{I}} \sum_{{}_B v \in \mathscr{V}} \| T_{AB}(\boldsymbol{\xi}))_B v - T_{AB}(\boldsymbol{\xi}_i))_B v \|_2^2 \,, \tag{11}$$

where $\mathscr{V}$ is the set of backprojected 3D points $B\mathbf{v}$ in the frame $B$, $\mathscr{I} = \{0, 1, 2, 3, 4\}$ denotes the pyramid levels, $\boldsymbol{\xi}_0$ is the predicted pose from the pose network and the other $\boldsymbol{\xi}_i$ are the estimated poses at the final iteration of Gauss-Newton optimisations on the respective pyramid level. This formulation enables the network to learn both feature and uncertainty representations in an end-to-end fashion, without the need for a ground truth feature map or ground truth correspondences, and without requiring an explicit definition of the uncertainty model. We set the feature map channels to be 8. Note that the uncertainty is defined as a scalar value. We unroll the Gauss-Newton optimisation and train all the models together from scratch using ADAM [31] for 30 epochs, with a learning rate initialized at 0.0005 and reduced at epochs [5, 10, 20]. When combining the ICP residual, we do a further fine-tuning for 10 epochs.

### E. Implementation Details

Fig 3 shows the architecture of our two-view encoder which takes the input from a pair of RGB-D images and extracts spatio-temporal correlation information from that. It is constructed into a 4-level pyramid architecture, where each level outputs a higher-dimension information. The architecture is modified from [10], however, we do not perform an average operation to extract feature maps. Instead, we send the outputs to the feature encoder and the uncertainty encoder to estimate the feature and uncertainty maps.

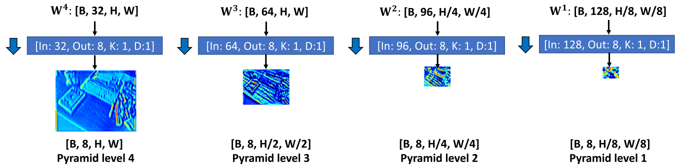

Fig. 4. The architecture of our feature encoder. On each pyramid level, it is a basic convolutional block that is group by a 1 by 1 convolutional layer, a BatchNorm layer, and a ELU layer. [In, Out, K, D] represents [Input channel, Output channel, Kernel size, Dilation] with stride always being 1.

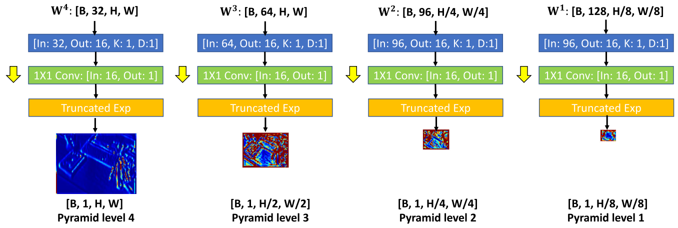

Fig. 5. The architecture of our uncertainty encoder. On each pyramid level, it is composed by a basic convolutional block, followed by a 1 by 1 convolutional layer and a truncated exponential operation.

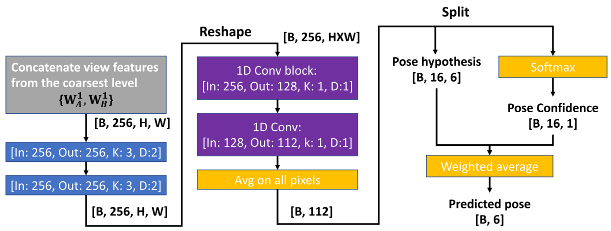

Fig. 6. The architecture of our pose network for initial pose prediction.

Fig 4 shows the architecture of our feature encoder on each pyramid level. It takes the input from the two-view encoder and predicts an 8-dimensional feature map.

Fig 5 shows the architecture of our uncertainty encoder on each pyramid level. It takes the input from the two-view encoder and predicts a 1-dimensional uncertainty map. We assume the output from the 1 by 1 convolutional layer is a logarithmised uncertainty and we use the exponentiation operation to recover the true uncertainty. The output is truncated to avoid gradient explosion.

Fig. 6 shows the architecture of our pose network to predict an initial pose on the coarsest level of the coarse-to-fine Gauss-Newton optimisation. It takes the input from a concatenation of the outputs of the two frames from the two-view encoder at the coarsest level. Similar to [23], the initial pose network also predicts multiple pose hypotheses and then fuse them together using their respective confidences. Here, we choose the hypotheses number to be 16. The pose is parameterised with 3 Euler angles and a 3-dimensional translation vector.

## IV. Experiments

### A. *Quantitative Evaluation and Discussion*

We first evaluate our method on the **TUM RGB-D** SLAM dataset [16]. A natural extension is to apply it to 3D rigid object motion estimation, which we test on the **MovingObjects3D** dataset [10].

**DeepIC** [10] is chosen as our main baseline method, which learns dense feature map for pose optimisation. To have a fair comparison, we use the same experimental setting as theirs. We randomly subsampled frames $B$ at intervals {1,2,4,8} relative to frame $A$ from TUM RGB-D dataset [16] and {1,2,4} from MovingObjects3D dataset [10] to generate various motion magnitudes and tracking difficulties as the training pairs. A comparison to this approach would show the importance of the uncertainty prediction and the initial pose prediction in our proposed method. We further implemented **DeepIC+P**, an augmented variant of DeepIC [10], with our pose prediction network to initialise their optimisation. The same number of iterations and pyramid levels are used as in our method. A comparison to it would further verify the contribution of our proposed probabilistic feature-metric loss.

To have a comparison to deep pose prediction methods that directly predict a relative transformation from two views, we implemented a **coarse-to-fine PoseNet**, similar to the tracking part in DeepTAM [23]. It is implemented on four pyramid levels for coarse-to-fine pose refinements, where the predicted pose from a coarser pyramid level would be used to bootstrap the prediction on a finer level. The network architecture is similar to our pose network but with different weights on different pyramid levels. A comparison to it would show a benefit of our learning-based optimisation approach for pose estimation. We further included the iterative refinement idea from [24] to the coarse-to-fine PoseNet approach. The **iterative PoseNet** has 3 iteration refinements on each pyramid level. All the learning-based comparison approaches are trained end-to-end using the loss in Eq. 11.

For the non-learning approaches, we compare our method to the pure geometric Point-to-Plane **ICP** method [30], which is essentially robust to illumination changes. We also include an **RGB-D VO** method [32] in the camera motion evaluation. A comparison to these approaches would show benefits of learning-based approaches, in terms of larger convergence basin and better accuracy, even under challenging lighting conditions.

To reveal the contribution of each component, we provide a detailed ablation study. We denote our system component, dense feature map, dense uncertainty map, deep initial pose prediction as F, U, P, respectively. We select the following settings. **Ours (F)**: We replace the uncertainty prediction with an identity uncertainty and disable the pose prediction with an identity pose initialisation. **Ours (F+P)**: We replace the uncertainty prediction with an identity uncertainty. **Ours (F+U)**: We disable the pose prediction and only use the proposed probabilistic feature-metric residual for alignment. **Ours (F+U+P)**: A full version of our probabilistic feature-metric tracking system. **Ours+ICP**: A combination of the probabilistic feature-metric and ICP residuals. All these combinations are implemented in coarse-to-fine optimisations, with the same number of iterations and pyramid levels as in the proposed method.The evaluation metrics are the 3D EPE loss in Eq. 11 and the relative pose error (RPE) metrics defined in TUM RGB-D dataset [16].

TABLE I
RESULTS ON OUR TEST SPLIT IN TUM RGB-D DATASET. KF DENOTES THE FRAME INTERVALS

| Method | 3D EPE (cm) / RPE translation (cm) / RPE rotation (Deg) | | | |
|---|---|---|---|---|
| | KF 1 | KF 2 | KF 4 | KF 8 |
| ICP [30] | 2.53/1.25/0.75 | 5.12/2.57/1.47 | 13.21/5.73/3.70 | 28.80/10.54/7.89 |
| RGB-D VO [32] | 2.31/1.03/0.55 | 4.38/2.81/1.39 | 12.67/5.95/3.99 | 31.13/13.83/9.20 |
| Coarse-to-fine PoseNet [23] | 1.88/1.91/0.80 | 3.08/3.76/1.42 | 5.82/7.30/2.76 | 15.43/13.16/5.73 |
| Iterative PoseNet [23], [24] | 1.76/1.86/0.84 | 2.70/3.61/1.53 | 4.75/7.28/2.73 | 12.74/13.12/5.23 |
| DeepIC [10] | 1.31/0.69/0.45 | 1.57/1.14/0.63 | 2.53/2.09/1.10 | 11.03/5.88/3.76 |
| DeepIC+P, adapted from [10] | 1.26/0.69/0.44 | 1.46/1.13/0.60 | 2.32/2.68/1.10 | 8.20/5.06/3.73 |
| Ours (F) | 1.25/0.67/0.44 | 1.49/1.14/0.60 | 2.50/2.78/1.14 | 11.70/12.20/4.37 |
| Ours (F+P) | 1.24/0.65/0.44 | 1.42/1.04/0.57 | 2.04/2.06/0.81 | 7.35/6.71/2.89 |
| Ours (F+U) | 1.23/0.58/0.41 | 1.40/0.86/0.50 | 2.33/1.99/0.87 | 13.24/12.92/4.59 |
| Ours (F+U+P) | 1.23/0.57/**0.40** | 1.38/0.80/0.48 | **1.71/1.22/0.64** | 5.48/4.89/2.12 |
| Ours+ICP | **1.22/0.54/0.40** | **1.33/0.76/0.47** | 1.78/1.26/0.66 | **4.82/4.57/2.00** |

TABLE II
RESULTS ON OUR TEST SPLIT OF MOVINGOBJECTS3D DATASET

| Method | 3D EPE (cm) / RPE translation (cm) / RPE rotation (Deg) | | |
|---|---|---|---|
| | KF 1 | KF 2 | KF 4 |
| ICP [30] | 3.31/9.75/**2.74** | 9.63/19.72/8.31 | 19.98/41.40/16.64 |
| Coarse-to-fine PoseNet [23] | 2.62/10.10/4.14 | 5.01/20.19/8.29 | 9.63/38.96/16.02 |
| Iterative PoseNet [23], [24] | 2.55/10.08/4.14 | 4.96/20.16/8.28 | 9.60/38.91/16.00 |
| DeepIC [10] | 2.91/9.73/3.74 | 5.94/19.60/7.41 | 12.96/38.39/14.71 |
| DeepIC+P, adapted from [10] | 2.66/9.78/3.76 | 5.14/19.72/7.67 | 9.90/38.50/15.17 |
| Ours (F) | 2.52/9.34/3.57 | 5.04/18.90/7.26 | 10.49/37.19/14.39 |
| Ours (F+P) | 2.64/9.59/3.64 | 5.14/19.42/7.43 | 9.97/37.01/14.32 |
| Ours (F+U) | 2.20/8.62/3.43 | 4.53/17.90/7.19 | 9.86/36.18/14.50 |
| Ours (F+U+P) | 2.17/8.44/3.22 | 4.47/17.86/6.91 | 9.26/36.443/14.22 |
| Ours+ICP | **1.93/7.84**/2.93 | **4.12/16.94/6.29** | **8.93/35.39/13.14** |

**TUM RGB-D Dataset:** We use the same setting as DeepIC [10], where sequences 'fr1/360,' 'fr1/desk,' 'fr2/360,' and 'fr2/pioneer360' are used for testing and the remaining sequences are split into training (first 95% of each sequence) and validation (last 5%). Images are transformed to a resolution of 160×120, with depth values outside of 0.5 m to 5.0 m being ignored.

Table I summarises the results on the TUM RGB-D dataset. Our method outperforms all the other state-of-the-art learning-based approaches, as well as the non-learning RGB-D VO, and ICP methods, from small baselines to large baselines. Compared with all ablation variants, our full version (F+U+P) achieves the best performance. The addition of uncertainty estimation complements the high-dimensional feature-metric alignment to improve the tracking accuracy. The predicted initial pose further improves the accuracy by bringing the estimation close the correct minimum, especially in the large motion scenarios. After fine-tuning the probabilistic combination with ICP loss, it can be seen that the performance is further improved in most cases (except KF 4 where the performance drops a bit), showing the validity of the probabilistic combination.

We have further developed a prototype visual odometry system, where the camera pose is estimated by our proposed method. Despite being a pure frame-to-frame tracking system without components of keyframing and loop closure optimisations, drift caused by incremental misalignment qualitatively remains small. The qualitative results can be found in the supplementary video.

**MovingObjects3D Dataset:** MovingObjects3D dataset contains 6 different catogories of objects moving in front of the camera under various illumination changes. We follow the dataset setting, where the categories of 'boat' and 'motorbike' are used as the testing set and the other categories are split into training (first 95% sequences of each category) and validation (last 5%), to test tracking performance for unseen objects. For the non-learning-based ICP [30] approach, we provide ground truth object masks for them to test their optimal performances. For the learning-based approaches, we reply on those systems to distinguish the object motion from the background, given the ground truth object and camera motions. Table II reports the results, which again show the superior performance of our method and confirm the contribution of each proposed component.

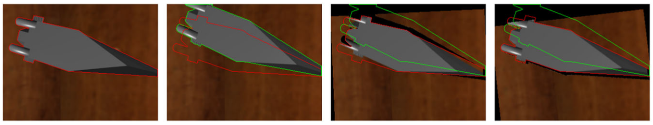
(a) Frame A (b) Frame B (c) Ours (d) Ours+ICP

Fig. 7. Qualitative results on MovingObjects3D dataset. Object motion between the frame A and frame B is estimated using our proposed method (c) and a further combination with ICP (d). The object is warped from frame A to B using the estimated motion for visualization. The ground truth object boundaries in $A$ and $B$ are colored in red and color, respectively. Black regions in the warped image are caused by occlusion.

Figure 7 visualises our tracking result on the test split of MovingObjects3D dataset. As can be seen, our proposed method can provide a good alignment for objects under large motion and lighting changes. A combination with ICP can provide a further refinement in the pose estimation.

**Ablation Study on the Choice of Channel Dimension:** As examined in [9], multi-dimensional feature map from network can improve tracking robustness. In Table I and II, **Ours (F)**,

TABLE III

ABLATION STUDY OF THE CHANNEL DIMENSION EFFECT ON OUR TEST SPLIT IN TUM RGB-D. F, U, C ABBREVIATE THE FEATURE MAP, UNCERTAINTY, AND THE CHANNEL DIMENSION. TIME IS THE AVERAGE INFERENCE TIME FOR A PAIR OF INPUT RGB-D IMAGES (SIZE 160×120)

| Map | C | 3D EPE (cm) / RPE translation (cm) / RPE rotation (Deg) | | | | Time |
|---|---|---|---|---|---|---|
| | | KF 1 | KF 2 | KF 4 | KF8 | (ms) |
| F<br>U=1 | 1 | 1.23/0.58/0.41 | 1.37/0.83/0.50 | 1.86/1.48/0.74 | 8.15/6.09/2.93 | 5.41 |
| | 3 | 1.23/**0.57/0.40** | 1.36/**0.78/0.48** | 1.72/1.24/0.64 | 5.92/5.05/2.20 | 6.25 |
| | 8 | 1.23/**0.57/0.40** | 1.38/0.80/**0.48** | 1.71/1.22/0.64 | **5.48/4.89/2.12** | 7.29 |
| | 16 | **1.22/0.57/0.40** | **1.35/0.78/0.48** | **1.66/1.21/0.62** | 5.72/4.94/2.22 | 11.67 |
| U<br>F=8 | 1 | 1.23/0.57/**0.40** | 1.38/0.80/**0.48** | **1.71/1.22/0.64** | **5.48/4.89/2.12** | 7.29 |
| | 8 | **1.22/0.55/0.40** | **1.37/0.79**/0.49 | 1.74/1.35/0.67 | 6.15/5.58/2.38 | 9.13 |

with higher-dimension features, outperforms [10] in most cases, even without uncertainty or pose predictions. On the other hand, a higher dimension of feature maps usually bring a higher computational cost. In this part, we experimentally evaluate the effect of the channel dimension of the feature map and the uncertainty map. We fix the uncertainty channel to be 1 when we vary the feature channels and fix the feature channel to be 8 when we vary the uncertainty channels between 1 and the same feature channel, i.e. 8. Table III summarises accuracy and inference time on the TUM RGB-D dataset [16]. Note that the accuracy increases when we increase the channel dimension of feature map, albeit with diminishing gains at dimensions higher than 8. When we increase the channel dimension of the uncertainty map, the accuracy very slightly increases for small baselines and slightly decreases for large baselines, validating the original choice of scalar uncertainty prediction.

In addition to accuracy, the increase of channel dimension in either feature or uncertainty map dimension would increase the GPU memory usage and reduce the inference speed. As a compromise of all these factors, we choose the feature dimension to be 8 and the uncertainty dimension to be 1 in all our other experiments.

**Model Size and Computation Time:** Our system implemented in PyTorch has 1.83 M learnable parameters. The average forward inference time for a pair of RGB-D image in the resolution of 160×120 on a GTX 1080 platform is 7.29 ms. After integrating ICP, it is 9.84 ms (i.e. +35%) on the same platform.

We also studied the effect of the input image resolution. With increased resolution (256×192), accuracy slightly improves on the small baselines, i.e. KF 1 and 2, however, slightly deteriorates on KF 4 and 8 while the computation increases to 15.29 ms (i.e. +111%). Therefore, we set 160×120 as main setting for training and testing.

### B. Qualitative Evaluation and Discussions

**Convergence Basin:** To analyse the effect of the initial pose prediction in our system, we perform a cost landscape visualisation experiment. Since $\xi$ is a 6D vector, it is computationally infeasible to sample cost on all possible pose components and also difficult to visualise the 6D cost landscape. Therefore, we choose to fix the rotation and z-translation components and only sample the pose combinations at the x and y translations around the ground truth pose. Fig. 8 shows one example on our test split from the TUM RGB-D dataset using the an interval of 8 frames. It can be seen that our pose prediction network brings the estimation into the convergence basin near the global minimum otherwise the conventional identity pose initialisation would lead the optimisation to a wrong local minimum.

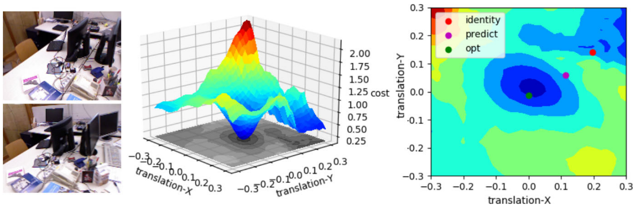

Fig. 8. Visualisation of cost landscape of x and y translation for the feature-metric loss on the coarsest level. From left to right: input, cost landscape 3D, and 2D projection of cost landscape.

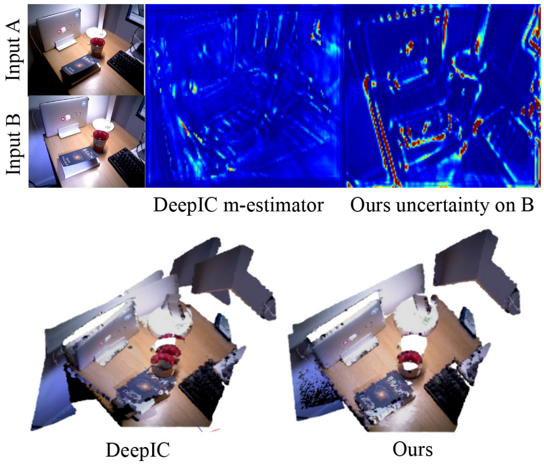

Fig. 9. Qualitative evaluation in challenging lighting. Notice our uncertainty estimation is more sensitive to the lighting changes than the learned m-estimator in DeepIC (higher value is in red and lower value is in blue).

**Challenging Illuminations:** Uncertainty prediction is significant for deploying neural network on robotic applications. DeepIC [10] proposed a learned robust cost function m-estimator to downweigh the residual outliers. To evaluate our learned uncertainty and also to compare to DeepIC's learned m-estimator, we captured sequences using an RGB-D camera while we were waving a flashlight to create illumination changes. The collected sequences contain both local and global lighting, reflection, and shading variances across the images. Since we don't have ground truth poses on these frames, we warp the point cloud from one frame to another using the estimated transformation between them and visualise the 3D pointcloud alignment of the two views. We test it using the weights trained from the TUM RGB-D dataset without fine-tuning. Fig. 9 shows one example. It can be seen that our method provides more robust pose estimation under those lighting changes. This is partially because our estimated uncertainty can more reliabely capture illumination variance, e.g. on the book and desk surface, than DeepIC's m-estimator. Please refer to the supplementary video for more results and details.

## V. Conclusion

We presented a deep probabilistic feature-metric two-frame RGB-D tracking method by combining the power of deep learning for feature learning, uncertainty estimation and pose prediction in a learning-based optimisation framework. It enables our method compact and to outperform the state of the art methods on camera motion and rigid object motion estimation benchmarks. Challenging experiments have shown an accurate and robust performance under large motion and strong lighting change scenarios, which is significant and currently lacking, in real-world robotic applications. We further showcased how our proposed residual can easily be combined with commonly used ICP residual in practice. Continuing from here, we would like to explore how to better combine the probabilistic feature-metric residuals with other residuals. Also, we aim to apply our tracking method to full dense SLAM systems, including object-level and dynamic SLAM systems.

## Acknowledgment

We wish to thank Shuaifeng Zhi, Jan Czarnowski and Tristan Laidlow for fruitful discussions and the anonymous reviewers for their comments and advice.

## References

[1] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artificial Intell.*, 1981, pp. 674–679.

[2] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.

[3] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "MID-Fusion: Octree-based object-level multi-instance dynamic slam," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 5231–5237.

[4] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intell.*, vol. 17, pp. 185–203, 1981.

[5] R. A. Newcombe, S. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis*, 2011, pp. 2320–2327.

[6] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[7] L. Platinsky, A. J. Davison, and S. Leutenegger, "Monocular visual odometry: Sparse joint optimisation or dense alternation?" in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 5126–5133, 2017.

[8] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot. (T-RO)*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[9] J. Czarnowski, S. Leutenegger, and A. J. Davison, "Semantic texture for robust dense tracking," in *Proc. Int. Conf. Comput. Vis. Workshops*, pp. 860–868, 2017.

[10] Z. Lv, F. Dellaert, J. Rehg, and A. Geiger, "Taking a deeper look at the inverse compositional algorithm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 4581–4590, 2019.

[11] C. Tang and P. Tan, "BA-Net: Dense bundle adjustment networks," in *Proc. Int. Conf. Learn. Representations*, 2019.

[12] T. Schmidt, R. Newcombe, and D. Fox, "Self-Supervised visual descriptor learning for dense correspondence," in *IEEE Robot. Autom. Letters,* vol. 2, no. 2, pp. 420–427, April, 2017.

[13] L. von Stumberg, P. Wenzel, Q. Khan, and D. Cremers, "GN-Net: The gauss-newton loss for multi-weather relocalization," in *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, 2020, pp. 890–897.

[14] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Neural Inform. Process. Syst.*, 2017.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[16] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[17] C. Jaramillo, Y. Taguchi, and C. Feng, "Direct multichannel tracking," in *Proc. Int. Conf. 3D Vis*, 2017, pp. 573–580.

[18] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1735–1742.

[19] C. Wang, H. K. Galoogahi, C.-H. Lin, and S. Lucey, "Deep-LK for efficient adaptive object tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 627–634, 2018.

[20] M. Bloesch, T. Laidlow, R. Clark, S. Leutenegger, and A. J. Davison, "Learning meshes for dense visual SLAM," in *Proc. Int. Conf. Comput. Vis.*, pp. 5855–5864, 2019.

[21] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6612–6619.

[22] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "DeMoN: Depth and motion network for learning monocular stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, 2017.

[23] H. Zhou, B. Ummenhofer, and T. Brox, "Deeptam: Deep tracking and mapping," in *Proc. Eur. Conf. Comput. Vis.*, pp. 822–838, 2018.

[24] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6d pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, pp. 683–698, 2018.

[25] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, pp. 7482–7491, 2018.

[26] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular slam with learned depth prediction," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2017, pp. 6565–6574.

[27] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, "Neural rgb(r)d sensing: Depth and uncertainty from a video camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 10986–10995, 2019.

[28] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, "D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1281–1292, 2020.

[29] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework: Part 1," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, 2004.

[30] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. IEEE Int. Workshop 3D Digital Imag. Model.*, 2001, pp. 145–152.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[32] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-Time Visual Odometry from Dense RGB-D Images," in *Proc. Workshop Live Dense Reconstruction Moving Cameras ICCV*, pp. 719–722, 2011.