Learning Accurate Dense Correspondences and When to Trust Them

Prune Truong Martin Danelljan Luc Van Gool Radu Timofte
Computer Vision Lab, ETH Zurich, Switzerland
{prune.truong, martin.danelljan, vangool, radu.timofte}@vision.ee.ethz.ch

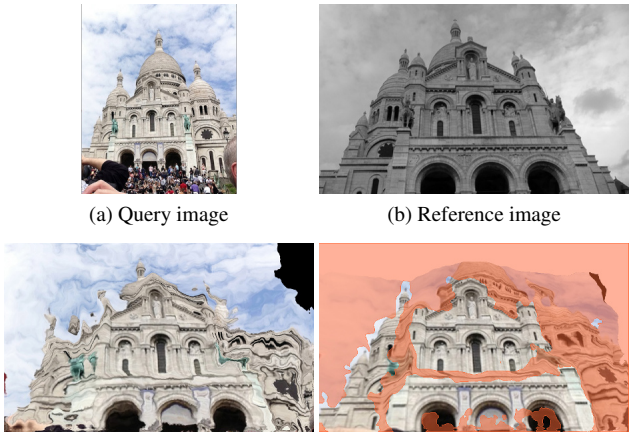
Abstract

Establishing dense correspondences between a pair of images is an important and general problem. However, dense flow estimation is often inaccurate in the case of large displacements or homogeneous regions. For most applications and down-stream tasks, such as pose estimation, image manipulation, or 3D reconstruction, it is crucial to know when and where *to trust the estimated matches.*

In this work, we aim to estimate a dense flow field relating two images, coupled with a robust pixel-wise confidence map indicating the reliability and accuracy of the prediction. We develop a flexible probabilistic approach that jointly learns the flow prediction and its uncertainty. In particular, we parametrize the predictive distribution as a constrained mixture model, ensuring better modelling of both accurate flow predictions and outliers. Moreover, we develop an architecture and training strategy tailored for robust and generalizable *uncertainty prediction in the context of self-supervised training. Our approach obtains state-of-the-art results on multiple challenging geometric matching and optical flow datasets. We further validate the usefulness of our probabilistic confidence estimation for the task of pose estimation. Code and models are available at* https://github.com/PruneTruong/PDCNet.

1. Introduction

Finding pixel-wise correspondences between pairs of images is a fundamental computer vision problem with numerous important applications, including dense 3D reconstruction [40], video analysis [33, 45], image registration [44, 50], image manipulation [11, 28], and texture or style transfer [20, 26]. Dense correspondence estimation has most commonly been addressed in the context of optical flow [2, 12, 16, 48], where the image pairs represent consecutive frames in a video. While these methods excel in the case of small appearance changes and limited displacements, they cannot cope with the challenges posed by the more general geometric matching task. In geometric matching, the images can stem from radically different views of the same scene, often captured by different cameras and at different occasions. This leads to large displacements and significant appearance transformations between the frames.



(a) Query image (b) Reference image (c) Baseline (d) **PDC-Net** (Ours)

Figure 1. Estimating dense correspondences between the query (a) and the reference (b) image. The query is warped according to the resulting flows (c)-(d). The baseline (c) does not estimate an uncertainty map and is therefore unable to filter the inaccurate flows at *e.g.* occluded and homogeneous regions. In contrast, our PDC-Net (d) not only estimates accurate correspondences, but also *when to trust them*. It predicts a robust uncertainty map that identifies accurate matches and excludes incorrect and unmatched pixels (red).

In contrast to optical flow, the more general dense correspondence problem has received much less attention [31, 37, 42, 52]. Dense flow estimation is prone to errors in the presence of large displacements, appearance changes, or homogeneous regions. It is also ill-defined in case of occlusions or in *e.g.* sky, where predictions are bound to be inaccurate (Fig. 1c). For geometric matching applications, it is thus crucial to know *when and where to trust* the estimated correspondences. For instance, pose estimation, 3D reconstruction, and image-based localization require a set of highly robust and accurate matches as input. The predicted dense flow field must therefore be paired with a *robust* confidence estimate (Fig. 1d). Uncertainty estimation is also indispensable for safety-critical tasks, such as autonomous driving and medical imaging. In this work, we set out to

expand the application domain of dense correspondence estimation by learning to predict reliable confidence values.

We propose the Probabilistic Dense Correspondence Network (PDC-Net), for joint learning of dense flow and uncertainty estimation, applicable even for extreme appearance and view-point changes. Our model predicts the conditional probability density of the flow, parametrized as a constrained mixture model. However, learning *reliable and generalizable* uncertainties without densely annotated real-world training data is a highly challenging problem. Standard self-supervised techniques [31, 35, 52] do not faithfully model real motion patterns, appearance changes, and occlusions. We tackle this challenge by introducing a carefully designed architecture and improved self-supervision to ensure robust and generalizable uncertainty predictions.

Contributions: Our main contributions are as follows. **(i)** We introduce a *constrained mixture model* of the predictive distribution, allowing the network to flexibly model both accurate predictions and outliers with large errors. **(ii)** We propose an architecture for predicting the parameters of our predictive distribution, that carefully exploits the information encoded in the correlation volume, to achieve generalizable uncertainties. **(iii)** We improve upon self-supervised data generation pipelines to ensure more robust uncertainty estimation. **(iv)** We utilize our uncertainty measure to address extreme view-point changes by iteratively refining the prediction. **(v)** We perform extensive experiments on a variety of datasets and tasks. In particular, our approach sets a new state-of-the-art on the Megadepth geometric matching dataset [25], on the KITTI-2015 training set [9], and outperforms previous dense methods for pose estimation on the YFCC100M dataset [49]. Moreover, without further post-processing, our confident dense matches can be directly input to 3D reconstruction pipelines [40], as shown in Fig. 2.

2. Related work

Confidence estimation in geometric matching: Only very few works have explored confidence estimation in the context of dense geometric or semantic matching. Novotny *et al*. [32] estimate the reliability of their trained descriptors by using a self-supervised probabilistic matching loss for the task of semantic matching. A few approaches [13, 36, 37] represent the final correspondences as a 4D correspondence volume, thus inherently encoding a confidence score for each tentative match. However, these approaches are usually restricted to low-resolution images, thus hindering accuracy. Moreover, generating one final confidence value for each match is highly non-trivial since multiple high-scoring alternatives often co-occur. Similarly, Wiles *et al*. [56] learn dense descriptors conditioned on an image pair, along with their distinctiveness score. However, the latter is trained with hand-crafted heuristics, while we instead do not make assumption on what the confidence score should be, and learn it directly from the data with a single unified loss. In DGC-Net, Melekhov *et al*. [31] predict both dense correspondence and matchability maps relating image pairs. However, their matchability map is only trained to identify out-of-view pixels rather than to reflect the actual reliability of the matches. Recently, Shen *et al*. [42] proposed RANSAC-Flow, a two-stage image alignment method, which also outputs a matchability map. It performs coarse alignment with multiple homographies using RANSAC on off-the-shelf deep features, followed by fine alignment. In contrast, we propose a unified network that estimates probabilistic uncertainties.

Figure 2. 3D reconstruction of Aachen [38] using the dense correspondences and uncertainties predicted by PDC-Net.

Uncertainty estimation in optical flow: While optical-flow has been a long-standing subject of active research, only a handful of methods provide uncertainty estimates. A few approaches [1, 3, 21, 22, 23] treat the uncertainty estimation as a post-processing step. Recently, some works propose probabilistic frameworks for joint optical flow and uncertainty prediction instead. They either estimate the model uncertainty [7, 17], termed epistemic uncertainty [19], or focus on the uncertainty from the observation noise, referred to as aleatoric uncertainty [19]. Following recent works [8, 58], we focus on aleatoric uncertainty and how to train a generalizable uncertainty estimate in the context of self-supervised training. Wannenwetsch *et al*. [55] propose ProbFlow, a probabilistic approach applicable to energy-based optical flow algorithms [3, 34, 46]. Gast *et al*. [8] propose probabilistic output layers that require only minimal changes to existing networks. Yin *et al*. [58] introduce HD^{3}F, a method to estimate uncertainty locally at multiple spatial scales and aggregate the results. While these approaches are carefully designed for optical flow data and restricted to small displacements, we consider the more general setting of estimating reliable confidence values for dense geometric matching, applicable to *e.g*. pose-estimation and 3D reconstruction. This brings additional challenges, including coping with significant appearance changes and large geometric transformations.

3. Our Approach: PDC-Net

We introduce PDC-Net, a method for estimating the dense flow field relating two images, coupled with a robust pixel-wise confidence map. The later indicates the reliability and accuracy of the flow prediction, which is necessary for pose estimation, image manipulation, and 3D-reconstruction tasks.

3.1. Probabilistic Flow Regression

We formulate dense correspondence estimation with a probabilistic model, which provides a framework for learning both the flow and its confidence in a unified formulation. For a given image pair $X = (I^q, I^r)$ of spatial size $H \times W$, the aim of dense matching is to estimate a flow field $Y \in \mathbb{R}^{H\times W\times 2}$ relating the reference I^r to the query I^q. Most learning-based methods address this problem by training a network F with parameters θ that directly predicts the flow as $Y = F(X;\theta)$. However, this does not provide any information about the confidence of the prediction.

Instead of generating a single flow prediction Y, our goal is to learn the conditional probability density $p(Y|X;\theta)$ of a flow Y given the input X. This is generally achieved by letting a network predict the parameters $\Phi(X;\theta)$ of a family of distributions $p(Y|X;\theta) = p(Y|\Phi(X;\theta)) = \prod_{ij} p(y_{ij}|\varphi_{ij}(X;\theta))$. To ensure a tractable estimation, conditional independence of the predictions at different spatial locations (i,j) is generally assumed. We use $y_{ij} \in \mathbb{R}^2$ and $\varphi_{ij} \in \mathbb{R}^n$ to denote the flow Y and predicted parameters Φ respectively, at the spatial location (i,j). In the following, we generally drop the sub-script ij to avoid clutter.

Compared to the direct approach $Y = F(X;\theta)$, the generated parameters $\Phi(X;\theta)$ of the predictive distribution can encode richer information about the flow prediction, including its uncertainty. In probabilistic regression techniques for optical flow [8, 17] and a variety of other tasks [19, 43, 54], this is most commonly performed by predicting the *variance* of the estimate y. In these cases, the predictive density $p(y|\varphi)$ is modeled using Gaussian or Laplace distributions. In the latter case, the density is given by,

$$\mathcal{L}(y|\mu,\sigma^2) = \frac{1}{\sqrt{2\sigma_u^2}} e^{-\sqrt{\frac{2}{\sigma_u^2}}|u-\mu_u|} . \frac{1}{\sqrt{2\sigma_v^2}} e^{-\sqrt{\frac{2}{\sigma_v^2}}|v-\mu_v|} \tag{1}$$

where the components u and v of the flow vector $y = (u,v) \in \mathbb{R}^2$ are modelled with two conditionally independent Laplace distributions. The mean $\mu = [\mu_u, \mu_v]^T \in \mathbb{R}^2$ and variance $\sigma^2 = [\sigma_u^2, \sigma_v^2]^T \in \mathbb{R}^2$ of the distribution $p(y|\varphi) = \mathcal{L}(y|\mu,\sigma^2)$ are predicted by the network as $(\mu,\sigma^2) = \varphi(X;\theta)$ at every spatial location.

3.2. Constrained Mixture Model Prediction

Fundamentally, the goal of probabilistic deep learning is to achieve a predictive model $p(y|X;\theta)$ that coincides with empirical probabilities as well as possible. We can get important insights into this problem by studying the empirical error distribution of a state-of-the-art matching model, in this case GLU-Net [52], as shown in Fig. 3. Errors can be categorized into two populations: inliers (in red) and outliers (in blue). Current probabilistic methods [8, 17, 33] mostly rely on a Laplacian model (1) of $p(y|X;\theta)$. Such a model is effective for correspondences which are easily estimated to be either inliers or outliers with *high certainty*, by predicting a low or high variance respectively. However, often the network is not certain whether a match is an inlier or outlier. A single Laplace can only predict an intermediate variance, which does not faithfully represent the more complicated uncertainty pattern in this case.

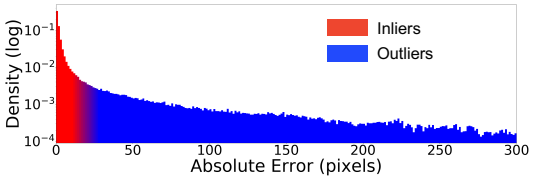


Figure 3. Distribution of errors $|\widehat{y}-y|$ on MegaDepth [25] between the flow $\widehat{y}$ estimated by GLU-Net [52] and the ground-truth y.

Mixture model: To achieve a flexible model capable of fitting more complex distributions, we parametrize $p(y|X;\theta)$ with a mixture model. In general, we consider a distribution consisting of M components,

$$p\left(y|\varphi\right) = \sum_{m=1}^{M} \alpha_m \mathcal{L}\left(y|\mu,\sigma_m^2\right) . \tag{2}$$

While we have here chosen Laplacian components (1), any simple density function can be used. The scalars $\alpha_m \geq 0$ control the weight of each component, satisfying $\sum_{m=1}^{M} \alpha_m = 1$. Note that all components have the same mean μ, which can thus be interpreted as the estimated flow vector, but different variances σ_m^2. The distribution (2) is therefore unimodal, but can capture more complex uncertainty patterns. In particular, it allows to predict the probability of inlier (red) and outlier (blue) matches (Fig. 3), each modeled by separate Laplace components.

Mixture constraints: In general, we now consider a network Φ that, for each pixel location, predicts the mean flow μ along with the variance σ_m^2 and weight α_m of each component, as $\left(\mu, (\alpha_m)_{m=1}^M, (\sigma_m^2)_{m=1}^M\right) = \varphi(X;\theta)$. However, a potential issue when predicting the parameters of a mixture model is its permutation invariance. That is, the predicted distribution (2) is unchanged even if we change the order of the individual components. This can cause confusion in the learning, since the network first needs to *decide* what each component should model before estimating the individual weights α_m and variances σ_m^2.

We propose a model that breaks the permutation invariance of the mixture (2), which simplifies the learning and

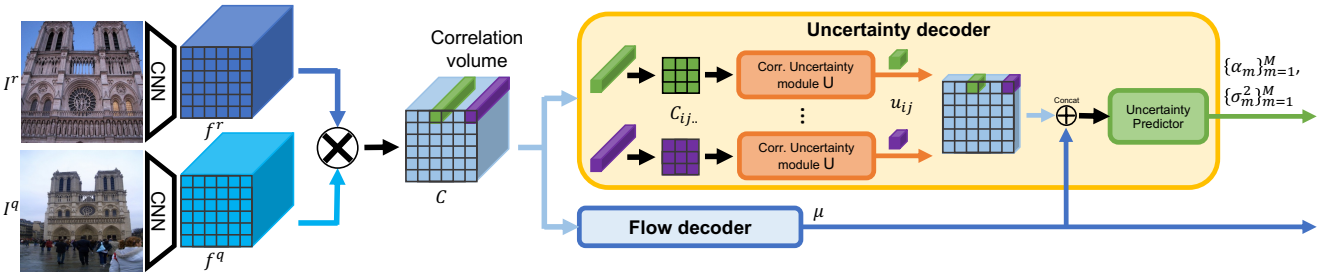


Figure 4. The proposed architecture for flow and uncertainty estimation. The correlation uncertainty module U_θ independently processes each 2D-slice $C_{ij..}$ of the correlation volume. Its output is combined with the estimated mean flow μ to predict the weight $\{\alpha_m\}_1^M$ and variance $\{\sigma_m^2\}_1^M$ parameters of our constrained mixture model (2)-(4).

greatly improves the robustness of the estimated uncertainties. In essence, each component m is tasked with modeling a specified range of variances σ_m^2. We achieve this by constraining the mixture (2) as,

$$0 < \beta_1^- \leq \sigma_1^2 \leq \beta_1^+ \leq \beta_2^- \leq \sigma_2^2 \leq \ldots \leq \sigma_M^2 \leq \beta_M^+ \quad (3)$$

For simplicity, we here assume a single variance parameter σ_m^2 for both the u and v directions in (1). The constants β_m^-, β_m^+ specify the range of variances σ_m^2. Intuitively, each component is thus responsible for a different range of uncertainties, roughly corresponding to different regions in the error distribution in Fig. 3. In particular, component $m = 1$ accounts for the most accurate predictions, while component $m = M$ models the largest errors and outliers. To enforce the constraint (3), we first predict an unconstrained value $h_m \in \mathbb{R}$, which is then mapped to the given range as,

$$\sigma_m^2 = \beta_m^- + (\beta_m^+ - \beta_m^-)\,\text{Sigmoid}(h_m). \quad (4)$$

The constraint values β_m^+, β_m^- can either be treated as hyper-parameters or learned end-to-end alongside θ.

Lastly, we emphasize an interesting interpretation of our constrained mixture formulation (2)-(3). Note that the predicted weights α_m, in practice obtained through a final SoftMax layer, represent the probabilities of each component m. Our network therefore effectively *classifies* the flow prediction at each pixel into the separate uncertainty intervals (3). We visualize in Fig. 5 the predictive log-distribution with $M = 2$ for three cases. The red and blue matches are with certainty predicted as inlier and outlier respectively, thus requiring only a single active component. In ambiguous cases (green), our mixture model (2)-(3) predicts the probability of inlier vs. outlier, giving a better fit compared to a single-component alternative. As detailed next, our network learns this ability without any extra supervision.

Training objective: As customary in probabilistic regression [5, 8, 10, 17, 19, 43, 53, 54], we train our method using the negative log-likelihood as the only objective. For one input image pair $X = (I^q, I^r)$ and corresponding ground-truth flow Y, the objective is given by

$$-\log p\big(Y|\Phi(X;\theta)\big) = -\sum_{ij} \log p\big(y_{ij}|\varphi_{ij}(X;\theta)\big)\,. \quad (5)$$

In Appendix B.1, we provide efficient analytic expressions of the loss (5) for our constrained mixture (2)-(4), that also ensure numerical stability. As detailed in Sec. 4.1, we can train our final model using either a self-supervised strategy where X and Y are generated by artificial warping, using real sparse ground-truth Y, or a combination of both. Next, we present the architecture of our network Φ that predicts the parameters of our constrained mixture model (2).

3.3. Uncertainty Prediction Architecture

Our aim is to predict an uncertainty value that quantifies the *reliability* of a proposed correspondence or flow vector. Crucially, the uncertainty prediction needs to *generalize* well to real scenarios, not seen during training. However, this is particularly challenging in the context of self-supervised training, which relies on synthetically warped images or animated data. Specifically, when trained on simple synthetic motion patterns, such as homography transformations, the network learns to heavily rely on global smoothness assumptions, which do not generalize well to more complex settings. As a result, the network learns to *confidently* interpolate and extrapolate the flow field to regions where no robust match can be found. Due to the significant distribution shift between training and test data, the network thus also infers confident, yet highly erroneous predictions in homogeneous regions on real data. In this section, we address this problem by carefully designing an architecture that greatly limits the risk of the aforementioned issues. Our architecture is visualized in Figure 4.

Current state-of-the-art dense matching architectures rely on feature correlation layers. Features f are extracted at resolution $h \times w$ from a pair of input images, and densely correlated either globally or within a local neighborhood of

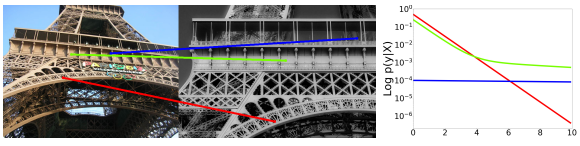
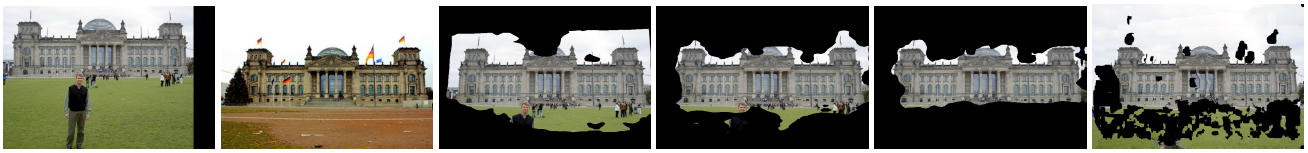


Figure 5. Predictive log-distr. $\log p(y|X)$ (2)-(3) for an inlier (red), outlier (blue), and ambiguous (green) match. Our mixture model faithfully represents the uncertainty also in the latter case.



(a) Query image (b) Reference image (c) Common decoder (d) Our decoder (e) Our decoder and data (f) RANSAC-Flow

Figure 6. Visualization of the estimated uncertainties by masking the warped query image to only show the confident flow predictions. The standard approach (c) uses a common decoder for both flow and uncertainty estimation. It generates overly confident predictions in the sky and grass. The uncertainty estimates are substantially improved in (d), when using the proposed architecture described in Sec. 3.3. Adding the flow perturbations for self-supervised training (Sec. 3.4) further improves the robustness and generalization of the uncertainties (e). For reference, we also visualize the flow and confidence mask (f) predicted by the recent state-of-the-art approach RANSAC-Flow [42].

size d. In the later case, the output correlation volume is best thought of as a 4D tensor $C \in \mathbb{R}^{h\times w\times d\times d}$. Computed as dense scalar products $C_{ijkl} = (f^r_{ij})^{\mathrm{T}} f^q_{i+k,j+l}$, it encodes the deep feature similarity between a location (i,j) in the reference frame I^r and a displaced location $(i+k, j+l)$ in the query I^q. Standard flow architectures process the correlation volume by first vectorizing the last two dimensions, before applying a sequence of convolutional layers over the *reference coordinates* (i,j) in order to predict the final flow.

Correlation uncertainty module: The straightforward strategy for implementing the parameter predictor $\Phi(X;\theta)$ is to simply increase the number of output channels to include all parameters of the predictive distribution. However, this allows the network to rely primarily on the local neighborhood when estimating the flow and confidence at location (i,j), and thus to ignore the actual reliability of the match and appearance information at the specific location. It results in over-smoothed and overly confident predictions, unable to identify ambiguous and unreliable matching regions, such as the sky. This is visualized in Fig. 6c.

We instead design an architecture that assesses the uncertainty at a specific location (i,j), without relying on neighborhood information. We note that the 2D slice $C_{ij..} \in \mathbb{R}^{d\times d}$ encapsulates rich information about the matching ability of location (i,j), in the form of a confidence map. In particular, it encodes the distinctness, uniqueness, and existence of the correspondence. We therefore create a *correlation uncertainty decoder* U_θ that independently reasons about each correlation slice as $U_\theta(C_{ij..})$. In contrast to standard decoders, the convolutions are therefore applied over *the displacement dimensions* (k,l). Efficient parallel implementation is ensured by moving the first two dimensions of C to the batch dimension using a simple tensor reshape. Our strided convolutional layers then gradually decrease the size $d \times d$ of the displacement dimensions (k,l) until a single vector $u_{ij} = U_\theta(C_{ij..}) \in \mathbb{R}^n$ is achieved for each spatial coordinate (i,j) (see Fig. 4).

Uncertainty predictor: The cost volume does not capture uncertainty arising at motion boundaries, crucial for real data with independently moving objects. We thus additionally integrate predicted flow information in the estimation of its uncertainty. In practise, we concatenate the estimated mean flow μ with the output of the correlation uncertainty module U_θ, and process it with multiple convolution layers. It outputs all parameters of the mixture (2), except for the mean flow μ (see Fig. 4). As shown in Fig. 6d, our uncertainty decoder, comprised of the correlation uncertainty module and the uncertainty predictor, successfully masks out most of the inaccurate and unreliable matching regions.

3.4. Data for Self-supervised Uncertainty

While designing a suitable architecture greatly alleviates the uncertainty generalization issue, the network still tends to rely on global smoothness assumptions and interpolation, especially around object boundaries (see Fig. 6d). While this learned strategy indeed minimizes the Negative Log Likelihood loss (5) on self-supervised training samples, it does not generalize to real image pairs. In this section, we further tackle this problem from the data perspective in the context of self-supervised learning.

We aim at generating less predictable synthetic motion patterns than simple homography transformations, to prevent the network from primarily relying on interpolation. This forces the network to focus on the appearance of the image region in order to predict its motion and uncertainty. Given a base flow $\tilde{Y}$ relating $\tilde{I}^r$ to $\tilde{I}^q$ and representing a simple transformation such as a homography as in prior works [31, 35, 51, 52], we create a residual flow $\epsilon = \sum_i \varepsilon_i$, by adding small local perturbations ε_i. The query image $I^q = \tilde{I}^q$ is left unchanged while the reference I^r is generated by warping $\tilde{I}^r$ according to the residual flow ϵ. The final perturbed flow map Y between I^r and I^q is achieved by composing the base flow $\tilde{Y}$ with the residual flow ϵ.

An important benefit of introducing the perturbations ε is to teach the network to be uncertain in regions where it cannot identify them. Specifically, in homogeneous regions such as the sky, the perturbations do not change the appearance of the reference ($I^r \approx \tilde{I}^r$) and are therefore unnoticed by the network. However, since the perturbations break the global smoothness of the synthetic flow, the flow errors on those pixels will be higher. In order to decrease the loss (5), the network will thus need to estimate a larger uncertainty for these regions. We show the impact of introducing the flow perturbations for self-supervised learning in Fig. 6e.

3.5. Geometric Matching Inference

In real settings with extreme view-point changes, flow estimation is prone to failing. Our confidence estimate can be used to improve the robustness of matching networks to such cases. Particularly, our approach offers the opportunity to perform multi-stage flow estimation on challenging image pairs, without any additional network components.

Confidence value: From the predictive distribution $p(y|\varphi(X;\theta))$, we aim at extracting a single confidence value, encoding the reliability of the corresponding predicted flow vector μ. Previous probabilistic regression methods mostly rely on the variance as a confidence measure [8, 17, 19, 54]. However, we observe that the variance can be sensitive to outliers. Instead, we compute the probability P_R of the true flow being within a radius R of the estimated mean flow μ. This is expressed as,

$$P_R = P(|y-\mu| < R) = \int_{\{y\in\mathbb{R}^2:|y-\mu|<R\}} p(y|\varphi)dy. \quad (6)$$

Compared to the variance, the probability value P_R also provides a more interpretable measure of the uncertainty.

Multi-stage refinement strategy: For extreme view-point changes with large scale or perspective variations, it is particularly difficult to infer the correct motion field in a single network pass. While this is partially alleviated by multi-scale architectures, it remains a major challenge in geometric matching. Our approach allows to split the flow estimation process into two parts, the first estimating a simple transformation, which is then used as initialization to infer the final, more complex transformation.

One of the major benefits of our confidence estimation is the ability to *identify* the set of accurate matches from the densely estimated flow field. After a first forward network pass, these accurate correspondences can be used to estimate a coarse transformation relating the image pair, such as a homography transformation. A second forward pass can then be applied to the coarsely aligned image pair, and the final flow field is constructed as a composition of the fine flow and the homography transform. While previous works also use multi-stage refinement [35, 42], our approach is much simpler, applying the *same* network in both stages and benefiting from the internal confidence estimation.

4. Experimental results

We integrate our approach into a generic pyramidal correspondence network and perform comprehensive experiments on multiple geometric matching and optical flow datasets. We also show that our method can be used for various tasks, including pose estimation and dense 3D reconstruction. Further results, analysis, visualizations and implementation details are provided in the Appendix.

4.1. Implementation Details

We adopt the recent GLU-Net-GOCor [51, 52] as our base architecture. It consists in a four-level pyramidal network operating at two image resolutions and employing a VGG-16 network [4] pre-trained on ImageNet for feature extraction. At each level, we add our uncertainty decoder (Sec. 3.3) and propagate the uncertainty prediction to the next level. We model the probability distribution $p(y|\varphi)$ with a constrained mixture (Sec. 3.2) with $M = 2$ Laplace components, where the first is fixed to $\sigma_1^2 = \beta_1^- = \beta_1^+ = 1$ to represent the very accurate predictions, while the second models larger errors and outliers, as $2 = \beta_2^- \leq \sigma_2^2 \leq \beta_2^+$, where β_2^+ is set to the square of the training image size.

Our training consists of two stages. First, we follow the self-supervised training procedure of [51, 52]. Random homography transformations are applied to images compiled from different sources to ensure diversity. For better compatibility with real 3D scenes and moving objects, the data is further augmented with random independently moving objects from the COCO [27] dataset. We further apply our perturbation strategy described in Sec. 3.4. In the second stage, we extend the self-supervised data with real image pairs with sparse ground-truth correspondences from the MegaDepth dataset [25]. We additionally fine-tune the backbone feature extractor. For fair comparison, we also train a version of GLU-Net-GOCor, denoted GLU-Net-GOCor*, using the same settings and data.

For datasets with very extreme geometric transformations, we also report using a multi-scale strategy. In particular, we extend our two-stage refinement approach (Sec. 3.5) by resizing the reference image to different resolutions. The resulting image pairs are passed through the network and we fit a homography for each pair, using our predicted flow and uncertainty map. We select the homography with the highest percentage of inliers, and scale it to the images original resolutions. The original image pair is then coarsely aligned and from there we follow the same procedure, as explained in Sec. 3.5. We refer to this option as Multi Scale (MS).

4.2. Geometric Correspondences and Flow

We first evaluate our PDC-Net in terms of the quality of the predicted flow field.

Datasets and metrics: We evaluate on standard datasets with sparse ground-truth, namely the **RobotCar** [24, 30],

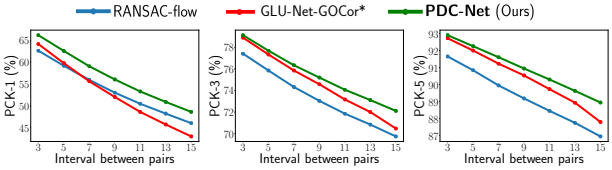


Figure 7. Results on ETH3D [41]. PCK-1 (left), PCK-3 (center) and PCK-5 (right) are plotted w.r.t. the inter-frame interval length.

	MegaDepth			RobotCar		
	PCK-1	PCK-3	PCK-5	PCK-1	PCK-3	PCK-5
SIFT-Flow [28]	8.70	12.19	13.30	1.12	8.13	16.45
NCNet [37]	1.98	14.47	32.80	0.81	7.13	16.93
DGC-Net [31]	3.55	20.33	32.28	1.19	9.35	20.17
GLU-Net [52]	21.58	52.18	61.78	2.30	17.15	33.87
GLU-Net-GOCor [51]	37.28	61.18	68.08	2.31	17.62	35.18
RANSAC-Flow (MS) [42]	53.47	83.45	86.81	2.10	16.07	31.66
GLU-Net-GOCor*	57.86	78.62	82.30	2.33	17.21	33.67
PDC-Net	70.75	86.51	88.00	2.54	**18.97**	**36.37**
PDC-Net (MS)	**71.81**	**89.36**	**91.18**	**2.58**	18.87	36.19

Table 1. PCK (%) results on sparse correspondences of the MegaDepth [25] and RobotCar [24, 30] datasets.

MegaDepth [25] and **ETH3D** [41] datasets. RobotCar depicts outdoor road scenes, taken under different weather and lighting conditions. Images are particularly challenging due to their numerous textureless regions. MegaDepth images show extreme view-point and appearance variations. Finally, ETH3D represents indoor and outdoor scenes captured from a moving hand-held camera. For RobotCar and MegaDepth, we evaluate on the correspondences provided by [42], which includes approximately 340M and 367K ground-truth matches respectively. For ETH3D, we follow the protocol of [52], sampling image pairs at different intervals to analyze varying magnitude of geometric transformations, resulting in 600K to 1100K matches per interval. In line with [42], we employ the Percentage of Correct Keypoints at a given pixel threshold T (PCK-T) as metric.

Results: In Tab. 1 we report results on MegaDepth and RobotCar. Our method PDC-Net outperforms all previous works by a large margin at all PCK thresholds. In particular, our approach is significantly more accurate and robust than the very recent RANSAC-Flow, which utilizes an extensive multi-scale (MS) search. Interestingly, our uncertainty-aware probabilistic approach also outperforms the baseline GLU-Net-GOCor* in pure flow accuracy. This clearly demonstrates the advantages of casting the flow estimation as a probabilistic regression problem, advantages which are not limited to uncertainty estimation. It also substantially benefits the accuracy of the flow itself through a more flexible loss formulation. In Fig. 7, we plot the PCKs on ETH3D. Our approach is consistently better than RANSAC-Flow and GLU-Net-GOCor* for all intervals.

Generalization to optical flow: We additionally show that our approach generalizes well to accurate estimation of optical flow, even though it is trained for the very different task of geometric matching. We use the established **KITTI** dataset [9], and evaluate according to the standard metrics, namely AEPE and F1. Since we do not fine-tune on KITTI, we show results on the training splits in Tab. 2. Our approach outperforms all previous generic matching methods (upper part) by a large margin in terms of both F1 and AEPE. Surprisingly, PDC-Net also obtains better results than all optical flow methods (bottom part), even outperforming the recent RAFT [48] on KITTI-2015.

	KITTI-2012		KITTI-2015	
	AEPE ↓	F1 (%) ↓	AEPE ↓	F1 (%) ↓
DGC-Net [31]	8.50	32.28	14.97	50.98
GLU-Net [52]	3.14	19.76	7.49	33.83
GLU-Net-GOCor [51]	2.68	15.43	6.68	27.57
RANSAC-Flow [42]	-	-	12.48	-
GLU-Net-GOCor*	2.26	10.23	5.58	18.76
PDC-Net	**2.08**	**7.98**	**5.22**	**15.13**
PWC-Net [47]	4.14	21.38	10.35	33.7
LiteFlowNet [14]	4.00	-	10.39	28.5
HD^{3}F [58]	4.65	-	13.17	24.0
LiteFlowNet2 [15]	3.42	-	8.97	25.9
VCN [57]	-	-	8.36	25.1
RAFT [48]	-	-	5.54	19.8

Table 2. Optical flow results on the training splits of KITTI [9]. The upper part contains generic matching networks, while the bottom part lists specialized optical flow methods, not trained on kitti.

4.3. Uncertainty Estimation

Next, we evaluate our uncertainty estimation. To assess the quality of the uncertainty estimates, we rely on Sparsification Error plots, in line with [1, 18, 55]. The pixels having the highest uncertainty are progressively removed and the AEPE or PCK of the remaining pixels is calculated, which results in the Sparsification curve. The Error curve is constructed by subtracting the Sparsification to the Oracle, for which the AEPE and PCK are calculated when the pixels are ranked according to the ground-truth error. As evaluation metric, we use the Area Under the Sparsification Error curve (AUSE). In Fig. 8, we compare the Sparsification Error plots on MegaDepth, of our PDC-Net with other dense methods providing a confidence estimation, namely DGC-Net [31] and RANSAC-Flow [42]. Our probabilistic method PDC-Net produces uncertainty maps that much better correspond to the true errors.

4.4. Pose and 3D Estimation

Finally, to show the joint performance of our flow and uncertainty prediction, we evaluate our approach for pose estimation. This application has traditionally been dominated by sparse matching methods.

Pose estimation: Given a pair of images showing different view-points of the same scene, two-view geometry estimation aims at recovering their relative pose. We follow the standard set-up of [59] and evaluate on 4 scenes of the **YFCC100M** dataset [49], each comprising 1000 image pairs. As evaluation metrics, we use mAP for dif-

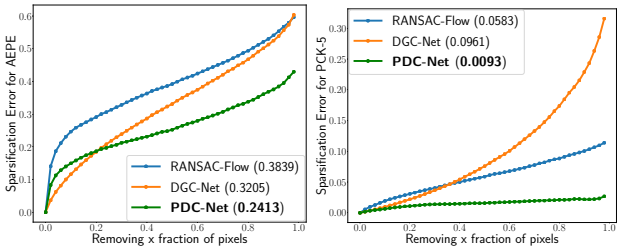


Figure 8. Sparsification Error plots for AEPE (left) and PCK-5 (right) on MegaDepth. Smaller AUSE (in parenthesis) is better.

ferent thresholds on the angular deviation between ground truth and predicted vectors for both rotation and translation. Results are presented in Tab. 3. Our approach PDC-Net outperforms the recent D2D [56] and obtains very similar results than RANSAC-Flow, while being 12.2 times faster. With our multi-scale (MS) strategy, PDC-Net outperforms RANSAC-Flow while being 3.6 times faster. Note that RANSAC-Flow employs its own MS strategy, using additional off-the-shelf features, which are exhaustively matched with nearest neighbor criteria [29]. In comparison, our proposed MS is a simpler, faster and more unified approach. We also note that RANSAC-Flow relies on a semantic segmentation network to better filter unreliable correspondences, in *e.g.* sky. Without this segmentation, the performance is drastically reduced. In contrast, our approach can directly estimate highly robust and generalizable confidence maps, without the need for additional network components. The confidence masks of RANSAC-Flow and our approach are visually compared in Fig. 6e-6f.

Extension to 3D reconstruction: We also qualitatively show the usability of our approach for dense 3D reconstruction. We compute dense correspondences between day-time images of the Aachen city from the Visual Localization benchmark [38, 39]. Accurate matches are then selected by thresholding our confidence map, and fed to COLMAP [40] to build a 3D point-cloud. It is visualized in Fig. 2.

4.5. Ablation study

Here, we perform a detailed analysis of our approach in Tab. 4. As baseline, we use a simplified version of GLU-Net, called BaseNet [52]. It is a three-level pyramidal network predicting the flow between an image pair. Our probabilistic approach integrated in this smaller architecture results in PDC-Net-s. All methods are trained using only the first-stage training, described in Sec. 4.1.

Probabilistic model (Tab. 4, top): We first compare BaseNet to our approach PDC-Net-s, modelling the flow distribution with a constrained mixture of Laplace (Sec. 3.2). On both KITTI-2015 and MegaDepth, our approach brings a significant improvement in terms of flow metrics. Note also that performing pose estimation by taking all correspondences (BaseNet) performs very poorly, which demonstrates the need for robust uncertainty estimation. While an unconstrained mixture of Laplace already drastically improves upon the single Laplace component,

	mAP @5°	mAP @10°	mAP @20°	Run-time (s)
Superpoint [6]	30.50	50.83	67.85	-
SIFT [29]	46.83	68.03	80.58	-
D2D [56]	55.58	66.79	-	-
RANSAC-Flow (MS+SegNet) [42]	64.88	73.31	81.56	9.06
RANSAC-Flow (MS) [42]	31.25	38.76	47.36	8.99
PDC-Net	63.98	73.48	81.91	**0.74**
PDC-Net (MS)	**65.20**	**74.51**	**83.04**	2.55

Table 3. Two-view geometry estimation on YFCC100M [49].

	KITTI-2015			MegaDepth			YFCC100M	
	AEPE	F1 (%)	AUSE	PCK-1 (%)	PCK-5 (%)	AUSE	mAP @5°	mAP @10°
BaseNet (L1-loss)	7.51	37.19	-	20.00	60.00	-	15.58	24.00
Single Laplace	6.86	34.27	0.220	27.45	62.24	0.210	26.95	37.10
Unconstrained Mixture	6.60	32.54	0.670	30.18	66.24	0.433	31.18	42.55
Constrained Mixture (PDC-Net-s)	**6.66**	**32.32**	**0.205**	**32.51**	**66.50**	**0.210**	**33.77**	**45.17**
Commun Dec.	6.41	32.03	**0.171**	31.93	**67.34**	0.213	31.13	42.21
Corr unc. module	**6.32**	**31.12**	0.418	31.97	66.80	0.278	**33.95**	**45.44**
Unc. Dec. (Fig 4) (PDC-Net-s)	6.66	32.32	0.205	**32.51**	66.50	**0.210**	33.77	45.17
BaseNet **w/o** Perturbations	7.21	37.35	-	20.74	59.35	-	15.15	23.88
BaseNet **w** Perturbations	7.51	37.19	-	20.00	60.00	-	15.58	24.00
PDC-Net-s **w/o** Perturbations	7.15	35.28	0.256	31.53	65.03	0.219	32.50	43.17
PDC-Net-s **w** Perturbations	**6.66**	**32.32**	**0.205**	**32.51**	**66.50**	**0.210**	**33.77**	**45.17**

Table 4. Ablation study. In the top part, different probabilistic models are compared (Sec. 3.1-3.2). In the middle part, a constrained Mixture is used, and different architectures for uncertainty estimation are compared (Sec. 3.3). In the bottom part, we analyze the impact of our training data with perturbations (Sec. 3.4).

the permutation invariance of the unconstrained mixture confuses the network, which results in poor uncertainty estimates (high AUSE). Constraining the mixture instead results in better metrics for both the flow and the uncertainty.

Uncertainty architecture (Tab. 4, middle): While the compared uncertainty decoder architectures achieve similar quality in flow prediction, they provide notable differences in uncertainty estimation. Only using the correlation uncertainty module leads to the best results on YFCC100M, since the module enables to efficiently discard unreliable matching regions, in particular compared to the common decoder approach. However, this module alone does not take into account motion boundaries. This leads to poor AUSE on KITTI-2015, which contains independently moving objects. Our final architecture (Fig. 4), additionally integrating the mean flow into the uncertainty estimation, offers the best compromise.

Perturbation data (Tab. 4, bottom): While introducing the perturbations does not help the flow prediction for BaseNet, it provides significant improvements in uncertainty *and* in flow performance for our PDC-Net-s. This emphasizes that the improvement of the uncertainty estimates originating from introducing the perturbations, also leads to improved and more generalizable flow predictions.

5. Conclusion

We propose a probabilistic deep network for estimating the dense image-to-image correspondences and associated confidence estimate. Specifically, we train the network to predict the parameters of the conditional probability density of the flow, which we model with a constrained mixture of Laplace distributions. Moreover, we introduce an architecture and improved self-supervised training strategy, designed for *robust and generalizable* uncertainty prediction. Our approach PDC-Net sets a new state-of-the-art on multiple geometric matching and optical flow datasets. It also outperforms dense matching methods on pose estimation.

Acknowledgements: This work was supported by the ETH Zürich Fund (OK), a Huawei Gift, Huawei Technologies Oy (Finland), Amazon AWS, and an Nvidia GPU grant.

References

[1] Oisin Mac Aodha, Ahmad Humayun, M. Pollefeys, and G. Brostow. Learning a confidence measure for optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1107–1120, 2013. 2, 7

[2] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011. 1

[3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 12:43–77, 1994. 2

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 6

[5] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 7181–7190. IEEE, 2020. 4

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 224–236, 2018. 8

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1050–1059. JMLR.org, 2016. 2

[8] Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3369–3378, 2018. 2, 3, 4, 6

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *I. J. Robotic Res.*, 32(11):1231–1237, 2013. 2, 7

[10] Fredrik K. Gustafsson, Martin Danelljan, Goutam Bhat, and Thomas B. Schön. Energy-based models for deep probabilistic regression. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XX*, volume 12365 of *Lecture Notes in Computer Science*, pages 325–343. Springer, 2020. 4

[11] Yoav HaCohen, Eli Shechtman, Dan B. Goldman, and Dani Lischinski. Non-rigid dense correspondence with applications for image enhancement. *ACM Trans. Graph.*, 30(4):70, 2011. 1

[12] Berthold K. P. Horn and Brian G. Schunck. "determining optical flow": A retrospective. *Artif. Intell.*, 59(1-2):81–87, 1993. 1

[13] Shuaiyi Huang, Qiuyue Wang, Songyang Zhang, Shipeng Yan, and Xuming He. Dynamic context correspondence network for semantic alignment. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2010–2019. IEEE, 2019. 2

[14] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8981–8989, 2018. 7

[15] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. A Lightweight Optical Flow CNN - Revisiting Data Fidelity and Regularization. 2020. 7

[16] Junhwa Hur and S. Roth. Optical flow estimation in the deep learning age. *ArXiv*, abs/2004.02853, 2020. 1

[17] Eddy Ilg, Özgün Çiçek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, pages 677–693, 2018. 2, 3, 4, 6

[18] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1647–1655. IEEE Computer Society, 2017. 7

[19] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5574–5584. Curran Associates, Inc., 2017. 2, 3, 4, 6

[20] Seungryong Kim, Dongbo Min, Somi Jeong, Sunok Kim, Sangryul Jeon, and Kwanghoon Sohn. Semantic attribute matching networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12339–12348, 2019. 1

[21] Claudia Kondermann, Daniel Kondermann, Bernd Jähne, and Christoph S. Garbe. An adaptive confidence measure for optical flows based on linear subspace projections. In *Pattern Recognition, 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007, Proceedings*, pages 132–141, 2007. 2

[22] Claudia Kondermann, Rudolf Mester, and Christoph S. Garbe. A statistical confidence measure for optical flows. In *Computer Vision - ECCV 2008, 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part III*, pages 290–301, 2008. 2

[23] Jan Kybic and Claudia Nieuwenhuis. Bootstrap optical flow confidence and uncertainty measure. *Comput. Vis. Image Underst.*, 115(10):1449–1462, 2011. 2

[24] Måns Larsson, Erik Stenborg, Lars Hammarstrand, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. A cross-season correspondence dataset for robust semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9532–9542, 2019. 6, 7

[25] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2041–2050, 2018. 2, 3, 6, 7

[26] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *ACM Trans. Graph.*, 36(4), July 2017. 1

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 6

[28] Ce Liu, Jenny Yuen, and Antonio Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):978–994, 2011. 1, 7

[29] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004. 8

[30] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. 6, 7

[31] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. DGC-Net: Dense geometric correspondence network. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 1, 2, 5, 7

[32] David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Self-supervised learning of geometrically stable features through probabilistic introspection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018. 2

[33] Jianing Qian, Junyu Nan, Siddharth Ancha, Brian Okorn, and David Held. Robust instance tracking via uncertainty flow. *CoRR*, abs/2010.04367, 2020. 1, 3

[34] Jérôme Revaud, Philippe Weinzaepfel, Zaïd Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, pages 1164–1172. IEEE Computer Society, 2015. 2

[35] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 39–48, 2017. 2, 5, 6

[36] I. Rocco, R. Arandjelović, and J. Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *European Conference on Computer Vision*, 2020. 2

[37] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelovic, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 1658–1669, 2018. 1, 2, 7

[38] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomás Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8601–8610, 2018. 2, 8

[39] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *British Machine Vision Conference, BMVC 2012, Surrey, UK, September 3-7, 2012*, pages 1–12, 2012. 8

[40] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4104–4113, 2016. 1, 2, 8

[41] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2538–2547, 2017. 6, 7

[42] Xi Shen, François Darmon, Alexei A Efros, and Mathieu Aubry. Ransac-flow: generic two-stage image alignment. In *16th European Conference on Computer Vision*, 2020. 1, 2, 5, 6, 7, 8

[43] Yichen Shen, Zhilu Zhang, Mert R. Sabuncu, and Lin Sun. Learning the distribution: A unified distillation paradigm for fast uncertainty estimation in computer vision. *CoRR*, abs/2007.15857, 2020. 3, 4

[44] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Data-driven visual similarity for cross-domain image matching. *ACM Transaction of Graphics (TOG) (Proceedings of ACM SIGGRAPH ASIA)*, 30(6), 2011. 1

[45] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 568–576. Curran Associates, Inc., 2014. 1

[46] Deqing Sun, Stefan Roth, and Michael J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *Int. J. Comput. Vision*, 106(2):115–137, Jan. 2014. 2

[47] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8934–8943, 2018. 7

[48] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, pages 402–419, 2020. 1, 7

[49] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: the new data in multimedia research. *Commun. ACM*, 59(2):64–73, 2016. 2, 7, 8

[50] Prune Truong, Stefanos Apostolopoulos, Agata Mosinska, Samuel Stucky, Carlos Ciller, and Sandro De Zanet. Glampoints: Greedily learned accurate match points. *2019*

IEEE/CVF International Conference on Computer Vision (ICCV), pages 10731–10740, 2019. 1

[51] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. GOCor: Bringing globally optimized correspondence volumes into your neural network. In *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020. 5, 6, 7

[52] Prune Truong, Martin Danelljan, and Radu Timofte. GLU-Net: Global-local universal network for dense flow and correspondences. In *2020 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2020*, 2020. 1, 2, 3, 5, 6, 7, 8

[53] Ali Varamesh and Tinne Tuytelaars. Mixture dense regression for object detection and human pose estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 13083–13092, 2020. 4

[54] Stefanie Walz, Tobias Gruber, Werner Ritter, and Klaus Dietmayer. Uncertainty depth estimation with gated images for 3d reconstruction. In *23rd IEEE International Conference on Intelligent Transportation Systems, ITSC 2020, Rhodes, Greece, September 20-23, 2020*, pages 1–8. IEEE, 2020. 3, 4, 6

[55] Anne S. Wannenwetsch, Margret Keuper, and Stefan Roth. Probflow: Joint optical flow and uncertainty estimation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1182–1191, 2017. 2, 7

[56] Olivia Wiles, Sébastien Ehrhardt, and Andrew Zisserman. D2D: learning to find good correspondences for image matching and manipulation. *CoRR*, abs/2007.08480, 2020. 2, 8

[57] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 794–805. Curran Associates, Inc., 2019. 7

[58] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6044–6053, 2019. 2, 7

[59] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Hongen Liao, and Long Quan. Learning two-view correspondences and geometry using order-aware network. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 5844–5853, 2019. 7