Unifying Flow, Stereo and Depth Estimation

Haofei Xu, Jing Zhang, Senior Member, IEEE, Jianfei Cai, Fellow, IEEE, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, Fellow, IEEE, and Andreas Geiger

Abstract—We present a unified formulation and model for three motion and 3D perception tasks: optical flow, rectified stereo matching and unrectified stereo depth estimation from posed images. Unlike previous specialized architectures for each specific task, we formulate all three tasks as a unified dense correspondence matching problem, which can be solved with a single model by directly comparing feature similarities. Such a formulation calls for discriminative feature representations, which we achieve using a Transformer, in particular the cross-attention mechanism. We demonstrate that cross-attention enables integration of knowledge from another image via cross-view interactions, which greatly improves the quality of the extracted features. Our unified model naturally enables cross-task transfer since the model architecture and parameters are shared across tasks. We outperform RAFT with our unified model on the challenging Sintel dataset, and our final model that uses a few additional task-specific refinement steps outperforms or compares favorably to recent state-of-the-art methods on 10 popular flow, stereo and depth datasets, while being simpler and more efficient in terms of model design and inference speed.

Index Terms—Cross-attention, dense correspondence, depth, optical flow, stereo, transformer.

I. INTRODUCTION

Understanding the 3D scene structure and motion from a set of 2D images has been a long-standing goal of computer vision [1], [2]. It is the cornerstone of many real-world applications, such as reconstructing a 3D city from internet photos [3], action recognition with optical flow [4], augmented reality [5] and autonomous driving [6].

Classic approaches typically tackle these tasks by solving an energy minimization problem with optimization techniques. For example, the variational approach for optical flow [7], semi-global matching for stereo vision [8] and bundle adjustment for structure-from-motion [9]. Although significant progress has been made with classic methods, they often still struggle in challenging situations like textureless regions and thin structures.

The rapid advancement of deep learning [10] and large-scale datasets also enables direct feed-forward inference of geometry and motion using high-capacity deep neural networks. Different network architectures have been proposed for different tasks in the last few years (e.g., FlowNet [11] for optical flow and MVSNet [12] for multi-view stereo). Further development of network architectures has led to steady progress on geometry and motion tasks, and learning-based methods are currently dominating the leaderboards of popular benchmarks [6], [13], [14], [15].

However, existing works are largely driven by designing task-specific models to solve each task independently, and thus a large variety of network architectures [16], [17], [18], [19], [20], [21], [22] have been proposed to handle different tasks, ignoring the fact that many multi-view geometry and motion tasks are fundamentally related correspondence estimation problems. Such a task-specific design philosophy inevitably leads to lots of architectures to deal with, and additional complexities are introduced in model deployment or update for real-world applications. Besides, pretrained models for different tasks cannot be reused (e.g., transfer between tasks) when they are studied in isolation.

In this paper, we aim at developing a single unified model to solve three dense perception tasks: optical flow, rectified stereo matching and unrectified stereo depth estimation from posed images, as shown in Fig. 1, which are fundamental building blocks for motion (optical flow) and 3D (depth) understanding. To achieve this, we first identify the main obstacle that hinders previous models to be generally applicable. In particular, previous methods mostly encode the task-specific geometric inductive bias (e.g., the cost volume [17], [23] with different shapes) as intermediate components of the model and use subsequent convolutional networks for flow/disparity/depth regression. Since the geometric inductive bias is task-dependent (e.g., optical flow’s cost volume is typically based on 2D correlation [11], while stereo matching networks construct cost volume by 1D correlation [19] or feature concatenation [24]), this leads to task-specific convolutional architectures for post-processing the cost volume. Moreover, the type of convolutional networks can be quite different (2D [17], [19], 3D [24], [25] or ConvGRU [21], [26]), which introduces additional challenges in unifying these tasks under such a pipeline.

Our key insight is that these tasks can be unified in an explicit dense correspondence matching formulation, where they can
be solved by directly comparing feature similarities. Thus the task is reduced to learning strong task-agnostic feature representations for matching, for which we use a Transformer [27], in particular the cross-attention mechanism to achieve this. We demonstrate that cross-attention can integrate the knowledge from another view via cross-view interactions, which greatly improves the quality of the extracted features. In our method, the geometric inductive biases for each task are modeled with parameter-free task-specific matching layers at the final output, which not only introduces no task-specific learnable parameters, but also demonstrates that cost volume post-processing is not always necessary for geometry and motion estimation tasks once we have strong features. This is different from Perceiver IO [28] that directly regresses optical flow without considering any geometric inductive bias, which is less efficient in terms of model parameters (ours is $8 \times$ less) and inference speed (ours is $4 \times$ faster). It also differs from IIB [29] that injects the geometric inductive bias at the input, which makes subsequent network layers task-specific. Our formulation implicitly assumes the corresponding pixels are visible on both images and thus they can be matched by comparing feature similarities. To handle unmatchable (occluded and out-of-boundary) regions, we introduce a simple task-agnostic self-attention layer to propagate the high-quality predictions to unmatched regions by measuring feature self-similarity [30], [31].

Our unified model naturally enables cross-task transfer since each task uses exactly the same learnable parameters for feature extraction. For example, without any finetuning, a pretrained optical flow model can be directly used for the task of rectified stereo matching and unrectified stereo depth estimation. Moreover, when finetuning with the pretrained flow model as initialization, we not only enjoy faster training speed for stereo and depth, but also achieve better performance, as evidenced by our experiments (Table X).

Our unified model with only one task-agnostic hierarchical matching refinement outperforms RAFT [21] with 31 refinement steps on the challenging Sintel [13] dataset while running faster (Fig. 5 and Table III), demonstrating the effectiveness and efficiency of our method. Our final model that uses a few additional task-specific refinement steps outperforms or compares favorably to recent state-of-the-art methods on 10 popular flow/stereo/depth datasets (KITTI Flow [14], Sintel [13], Middlebury [15], KITTI Stereo [14], ETH3D Stereo [32], Argoverse Stereo [33], ScanNet [34], SUN3D [35], RGBD-SLAM [36] and Scenes11 [37]), while being simpler and more efficient in terms of model design and inference speed.

This work represents a substantial extension of our previous CVPR 2022 conference paper GMFlow [38], where the new contributions are summarized as follows: (1) The initial work GMFlow [38] aims at demonstrating a successful alternative to RAFT’s [21] iterative architecture for the optical flow task, while this work proposes a more holistic perspective that unifies three dense correspondence estimation tasks. (2) We extend GMFlow to rectified stereo matching and unrectified stereo depth estimation from posed images and conduct extensive experiments. (3) We study the cross-task transfer behavior by reusing pretrained models. Our project page is available at haofei.xu.github.io/unimatch, and our code and models are available at github.com/autonomousvision/unimatch.

II. RELATED WORK

Most existing methods for optical flow, rectified stereo matching and unrectified stereo depth estimation have been largely driven by designing specific architectures for each specific task, without pursing a unified model. In this section, we will first review the development of each task independently, and then discuss their relations from the perspective of a unified model and multi-task learning.

A. Optical Flow

Optical flow has been traditionally tackled with variational approaches [2], [7], [39], [40], [41], [42], where it is typically solved as an energy minimization problem that consists of a brightness constancy term and a regularization term. The advancement of deep learning has also enabled directly learning optical flow from data. The pioneering learning-based work, FlowNet [11], proposed a convolutional neural network that directly takes two images as input and regresses an optical flow field. Further advances of network architectures and training strategies [16], [17], [21], [31], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53] have led to steady progress for learning-based methods, which today outperform traditional approaches by a large margin and are currently dominating the benchmarks including Sintel [13] and KITTI [6], [14].

However, a closer look at existing learning-based approaches reveals that the underlying architectural principles haven’t changed much since FlowNet [11], that is, regressing optical flow from local correlation (i.e., cost volume) with convolutions. Such a local regression approach is intrinsically limited by trading off large-displacement flow estimation with the size of the cost volume. To alleviate this problem, two popular strategies are coarse-to-fine [17], [43] and iterative refinement [21], [44] methods, which estimate large displacements incrementally in multiple stages. However, coarse-to-fine methods tend to miss
| Method                  | #blocks | Things (val, clean) | Sintel (train, clean) | Sintel (train, final) | Param (M) |
|------------------------|---------|---------------------|-----------------------|-----------------------|-----------|
|                        |         | EPE                 | EPE                   | EPE                   |           |
|                        |         | $50 - 10$ | $510 - 40$ | $840 +$ | $90 - 10$ | $910 - 40$ | $840 +$ |           |
| cost volume + conv     | 0       | 18.83, 3.42, 6.49 | 49.65, 6.45, 1.75 | 7.17, 38.19, 7.75 | 21.00, 8.88, 45.29 | 1.8 |
|                        | 4       | 10.99, 1.70, 3.41 | 29.78, 3.32, 0.73 | 3.84, 20.58, 4.93 | 0.99, 5.71, 31.16 | 4.6 |
|                        | 8       | 9.59, 1.44, 2.96 | 26.04, 2.89, 0.65 | 3.36, 17.79, 4.32 | 0.88, 4.95, 27.33 | 8.0 |
|                        | 12      | 9.04, 1.37, 2.84 | 24.46, 2.78, 0.65 | 3.32, 16.69, 4.07 | 0.84, 4.76, 25.44 | 11.5 |
|                        | 18      | 8.67, 1.33, 2.74 | 23.43, 2.61, 0.59 | 3.07, 15.91, 3.94 | 0.82, 4.62, 24.58 | 15.7 |
| Transformer + softmax  | 0       | 22.93, 8.57, 11.13 | 52.07, 8.44, 2.71 | 11.60, 42.10, 10.28 | 3.11, 13.83, 53.34 | 1.0 |
|                        | 1       | 11.45, 2.98, 4.68 | 28.35, 4.12, 1.27 | 5.08, 22.25, 6.11 | 1.70, 7.89, 33.52 | 1.6 |
|                        | 2       | 8.59, 1.80, 3.28 | 21.99, 3.09, 0.90 | 3.66, 17.37, 4.54 | 1.24, 5.44, 26.00 | 2.1 |
|                        | 4       | 7.19, 1.40, 2.62 | 18.66, 2.43, 0.67 | 2.73, 14.23, 3.78 | 1.01, 4.27, 22.37 | 3.1 |
|                        | 6       | 6.67, 1.26, 2.40 | 17.37, 2.28, 0.58 | 2.49, 13.89, 3.44 | 0.80, 3.97, 21.02 | 4.2 |
| conv + softmax         | 6       | 17.06, 5.79, 7.74 | 40.03, 6.36, 2.15 | 8.53, 31.53, 8.00 | 2.45, 10.42, 42.09 | 5.1 |

We stack different numbers of convolutional residual blocks or Transformer blocks to see how performance varies. All models are trained on Chairs and Things training sets. We report the performance on Things (clean) validation set and cross-dataset generalization results on Sintel (clean and final) training sets. Our method outperforms previous cost volume and convolution-based approaches by a large margin, especially for large motions ($s_k^+$). Replacing the Transformer in our model with a convolutional network (i.e., conv + softmax) leads to a significant performance drop, since convolutions are not able to model cross-view interactions (which is important for obtaining high-quality discriminative features, see also Table IIa).

**TABLE II**

| setup                  | Things (val, clean) clean | Sintel (train, final) clean | Param (M) |
|------------------------|---------------------------|-----------------------------|-----------|
|                        | EPE $50 - 10$ | $510 - 40$ | $840 +$ | $90 - 10$ | $910 - 40$ | $840 +$ |           |
| full                   | 6.67, 2.28, 3.44 | 4.2 |
| w/o cross attn.        | 10.84, 4.48, 6.32 | 3.8 |
| w/o position           | 9.18, 2.85, 4.28 | 4.2 |
| w/o FFN                | 8.71, 3.10, 4.43 | 1.8 |
| w/o self attn.         | 7.08, 2.49, 3.69 | 3.8 |

(a) **Transformer components.** Cross-attention contributes most.

(b) **Numbers of window splits in shifted local attention.** $2 \times 2$ represents a good speed-accuracy trade-off.

(c) **Global vs. local matching.** Global matching is significantly better for large motions while being fast to compute.

(d) **Flow propagation.** Greatly improves unmatched pixels.

All models are trained on Chairs and Things training sets.

**TABLE III**

| Method                  | #refine (RAFT) | Things (val, clean) | Sintel (train, clean) | Sintel (train, final) | Param (M) | Time (ms) |
|------------------------|----------------|---------------------|-----------------------|-----------------------|-----------|-----------|
|                        | EPE $50 - 10$ | $510 - 40$ | $840 +$ | EPE $90 - 10$ | $910 - 40$ | $840 +$ |           |
| RAFT [21]              | 0              | 14.28, 1.47, 3.62 | 40.48, 4.04, 0.77 | 4.30, 26.66, 5.45 | 0.99, 6.30 | 35.19 | 25 (14) |
|                        | 3              | 6.27, 0.69, 1.67 | 17.63, 1.92, 0.47 | 2.32, 11.37, 3.25 | 0.65, 20.04 |           | 39 (21) |
|                        | 7              | 4.66, 0.55, 1.38 | 12.87, 1.61, 0.39 | 1.90, 9.61, 2.80 | 0.53, 3.30 | 17.76 | 58 (31) |
|                        | 11             | 4.31, 0.53, 1.33 | 11.29, 1.55, 0.41 | 1.73, 9.19, 2.72 | 0.52, 3.12 | 17.43 | 78 (41) |
|                        | 23             | 4.22, 0.53, 1.32 | 11.52, 1.47, 0.36 | 1.63, 9.00, 2.69 | 0.52, 3.05 | 17.28 | 133 (71) |
|                        | 31             | 4.25, 0.53, 1.31 | 11.63, 1.41, 0.32 | 1.55, 8.83, 2.59 | 0.52, 3.00 | 17.45 | 170 (91) |
| GMFlow                 | 0              | 3.48, 0.67, 1.31 | 8.97, 1.50, 0.46 | 1.77, 8.26, 2.96 | 0.72, 3.45 | 17.70 | 57 (26) |
|                        | 1              | 2.80, 0.53, 1.01 | 7.31, 1.08, 0.30 | 1.25, 6.26, 2.48 | 0.51, 2.81 | 15.67 | 47 (15) |

The models are trained on Chairs and Things training sets. The inference time is measured on a single V100 and A100 (in parentheses) GPU at Sintel resolution ($436 \times 1024$). Our method gains more speedup than RAFT ($2.29 \times$ vs. $1.87 \times$), i.e., ours: 151 → 66, RAFT: 170 → 91) on the high-end A100 GPU since our method doesn’t require a large number of sequential computations.
fast-moving small objects if the resolution is too coarse and may suffer from the error-propagation issue [54]. In contrast, the iterative approaches like RAFT [21] lead to a linear increase in processing time due to the large number of sequential refinements. In contrast, we reformulate optical flow as a global matching problem, which identifies dense correspondences by directly comparing pair-wise feature similarities, leading to significant improvement for large displacements.

B. Stereo Matching

Typical stereo matching methods generally follow a four-step pipeline [23]: matching cost computation, cost aggregation, disparity computation and disparity refinement. Again, early optimization-based methods [55], [56] have been replaced by modern deep learning-based approaches [19], [24], [25], [57], [58], [59], [60], [61]. The current representative stereo methods can be broadly classified into two categories: 3D and 2D convolution-based approaches. Their key difference lies in the cost volume construction method. 3D convolution-based methods [22], [24], [25], [60] typically use feature concatenation while 2D methods [19], [58] use feature correlation. These methods usually build a local cost volume with a predefined search space (typically 192 pixels [59]) and the final disparity prediction is obtained by computing the weighted sum of all disparity candidates. Thus the output is always constrained by the predefined disparity range, which makes these methods less flexible to handle unconstrained settings like high-resolution images or new camera settings. For example, to adapt such an architecture to larger disparity ranges, the full model has to be re-trained by setting a new predefined maximum disparity. In contrast, we directly perform global matching along the scanline, which make no assumption on the disparity range and is able to handle arbitrary image resolutions.

Recent iterative 2D methods like RAFT-Stereo [26] and CREStereo [62] mostly follow the high-level design of the RAFT [21] architecture for optical flow, while introducing several task-specific components (e.g., 1D correlation) to make such a method suitable for the stereo matching task. In contrast, we show that our matching-based perspective enables to use the same model for both optical flow and stereo matching, with exactly the same learnable parameters. Besides, our model is also more efficient since we don’t rely on any 3D convolutions or a large number of sequential refinements. On the other hand, although MC-CNN [57] also tries to learn strong features for matching, the features in MC-CNN are extracted independently with a convolutional network, without considering cross-view interactions. However, as evidenced by our results, cross-view interactions are crucial for strong and discriminative features (see Tables I and II(a)).

Perhaps the most related stereo work to ours is STTR [63], which also uses a Transformer and matching-based disparity computation. However, STTR relies on a complex optimal transport matching layer and doesn’t produce predictions for occluded pixels, while we use a much simpler softmax operation and a simple flow propagation layer to handle occlusions. The later CSTR (Context-Enhanced Stereo Transformer) [64] tries to improve STTR’s performance with a new Transformer architecture, but it still suffers from the limitation of STTR. Moreover, STTR is designed to solve the stereo matching task, while we are seeking a unified model applicable to three different dense correspondence estimation tasks.

C. Depth Estimation

Learning-based depth estimation methods can be broadly categorized into monocular and multi-view approaches. Monocular methods [65], [66], [67], [68], [69], [70] take a single image as input and use generic network architectures like ResNet [71] to predict the dense depth map, while multi-view methods [12], [20], [37], [72], [73], [74], [75], [76], [77] usually focus on how to encode the geometric inductive bias (cost volume, warping, etc.) into the network architecture. Compared with monocular methods, multi-view depth estimation can better leverage the information from additional viewpoints and usually lead to improved performance [78]. Since multi-view information (e.g., video sequences) are usually readily available for many applications, we consider multi-view depth estimation in this paper. A popular multi-view depth pipeline is using the plane-sweep stereo [72], [79] approach, where different depth planes are tested for correctness. However, like rectified stereo matching, the state-of-the-art methods are usually dominated by 3D convolution-based approaches [72], [73], which accordingly introduces cubic computational complexity. In this paper, we approach this task from an explicit matching-based perspective and use a Transformer to obtain strong features for matching, achieving highly competitive performance without relying on any 3D convolutions. This is different from the recent work TransMVSNet [80], which still relies on 3D convolutions for cost volume post-processing and where the Transformer is used before the cost volume construction stage. Thus, our method is simpler and more lightweight.

D. Unified Model

Unified models aim at using task-agnostic architectures to solve different tasks. One notable work is Perceiver IO [28], which proposes a general Transformer architecture for different problems in different domains. Perceiver IO has been applied to the optical flow task, where a direct concatenation of two input images is fed to the Transformer, and optical flow is regressed without using any inductive bias. Despite its architectural simplicity, more parameters (8× more than ours) and additional computational complexity (4× slower than ours) are introduced in order to make the model perform well. Perceiver IO has also been used to solve the unrectified stereo depth estimation task [29], where the geometric inductive bias is fed into the network as additional inputs. Different from Perceiver IO, our design is motivated from a unified perspective that learns strong feature representations for dense correspondence matching for geometry and motion tasks. In our method, the geometric inductive biases are well-preserved at the final parameter-free matching layers, which doesn’t introduce any task-specific learnable parameters. This is different from Perceiver IO for optical flow and stereo depth estimation, where the network inputs are task-specific and thus it is not easy to reuse the model parameters from different tasks. Another related work is HD3 [18], which proposes a model that is applicable to both optical flow and stereo matching. However, HD3 relies on task-specific correlations.
(2D or 1D) as intermediate network components, resulting to task-specific learnable parameters in the subsequent decoders and thus making it not easy to transfer pretrained models across tasks.

III. METHODOLOGY

Dense correspondences between different viewpoints are the core of optical flow, rectified stereo matching and unrectified stereo depth estimation tasks. To unify these three tasks, our key idea is to use an explicit dense correspondence matching formulation, which identifies the solution by directly comparing feature similarities. Such a formulation calls for discriminative features, for which we use a Transformer, in particular the cross-attention to achieve this. The cross-attention can integrate the knowledge from another image via cross-view interactions, which greatly improves features’ quality and is not achievable with convolutions that operate on each view independently [57].

Fig. 2 provides an overview of our proposed method. We first extract dense features from two input images and then obtain the prediction with a parameter-free matching layer. A final self-attention layer is used to propagate the high-quality predictions to unmatched regions by measuring feature self-similarity.

The matching layers are parameter-free since they only compare feature similarities. The learnable parameters of our matching layers for optical flow, rectified stereo matching and unrectified stereo depth estimation under our unified matching-based formulation.

1) Flow Matching: Optical flow represents the apparent motion between two video frames, which can be computed by finding 2D pixel-wise dense correspondences on the image plane. To achieve this, we directly compare the feature similarities for each location in \( F_1 \) with respect to all locations in \( F_2 \) by computing their correlations (i.e., global matching). This can be implemented efficiently using a simple matrix multiplication:

\[
C_{\text{flow}} = \frac{F_1 F_2^T}{\sqrt{D}} \in \mathbb{R}^{H \times W \times H \times W},
\]

where each element in the correlation matrix \( C_{\text{flow}} \) represents the correlation value between coordinates \( p_1 = (i,j) \) in \( F_1 \) and \( p_2 = (k,l) \) in \( F_2 \), and \( \frac{1}{\sqrt{D}} \) is a normalization factor to avoid large values after the dot-product operation [27].

To obtain dense correspondences, we use a softmax matching layer [19], [59], [81], which is not only end-to-end differentiable but also enables sub-pixel accuracy. Specifically, we first normalize the last two dimensions of \( C_{\text{flow}} \) with the softmax operation, which gives us a distribution

\[
M_{\text{flow}} = \text{softmax}(C_{\text{flow}}) \in \mathbb{R}^{H \times W \times H \times W}
\]

for each position in \( F_1 \) with respect to all positions in \( F_2 \). Then, the correspondence \( G_{2D} \) can be obtained from the weighted average of the matching distribution \( M_{\text{flow}} \) with the 2D coordinates of pixel grid \( G_{2D} \in \mathbb{R}^{H \times W \times 2} \):

\[
G_{2D} = M_{\text{flow}} G_{2D} \in \mathbb{R}^{H \times W \times 2}.
\]

Finally, the optical flow \( V_{\text{flow}} \) can be obtained by computing the difference between the corresponding pixel coordinates:

\[
V_{\text{flow}} = G_{2D} - G_{2D} \in \mathbb{R}^{H \times W \times 2}.
\]
2) Stereo Matching: Rectified stereo matching aims to find the per-pixel disparity along the horizontal scanline (1D correspondence) between a rectified stereo pair, which can be viewed as a special case of 2D optical flow. Unlike the 2D global matching for optical flow in (1), we only need to consider matching along the 1D horizontal direction. More specifically, the correlation matrix for rectified stereo matching is

\[ C_{\text{disp}} \in \mathbb{R}^{H \times W \times W}. \]  

(5)

Similarly, we normalize the last dimension of \( C_{\text{disp}} \) and obtain the matching distribution

\[ M_{\text{disp}} = \text{softmax}(C_{\text{disp}}) \in \mathbb{R}^{H \times W \times W}. \]  

(6)

Considering that the correspondence of each pixel in the first image is located to the left of its reference pixel, we mask the upper triangle of the \( W \times W \) slices of \( M_{\text{disp}} \) to avoid unnecessary matches. Then, the 1D correspondence \( G_{1D} \in \mathbb{R}^{H \times W} \) can be obtained by computing the weighted average of the matching distribution \( M_{\text{disp}} \) with all potential horizontal locations \( P = [0, 1, 2, \ldots, W - 1] \in \mathbb{R}^W \):

\[ \hat{G}_{1D} = M_{\text{disp}} P \in \mathbb{R}^{H \times W}. \]  

(7)

Finally, the (positive) disparity can be obtained by computing the difference between the corresponding coordinates of the 1D horizontal pixel grid \( G_{1D} \in \mathbb{R}^{H \times W} \) (which stores only the \( x \)-coordinates) and \( \hat{G}_{1D} \):

\[ V_{\text{disp}} = G_{1D} - \hat{G}_{1D} \in \mathbb{R}^{H \times W}. \]  

(8)

3) Depth Matching: For unrestricted stereo depth estimation, we assume the camera intrinsic and extrinsic parameters \((K_1, E_1, K_2, E_2)\) for image \( I_1 \) and \( I_2 \) are known (i.e., poses). They can be obtained via additional sensors like IMU and GPS, or reliably estimated using Structure-from-Motion software like COLMAP [82]. To estimate depth, we take an approach similar to the classic plane-sweep stereo method [83]. More specifically, we first discretize a predefined depth range \([d_{\text{min}}, d_{\text{max}}]\) as \([d_1, d_2, \ldots, d_N]\) (in our implementation, we discretize the inverse depth domain, while we use depth here for ease of notation). Then for each depth candidate \( d_i (i = 1, 2, \ldots, N) \), we compute the 2D correspondences \( G_{2D} \in \mathbb{R}^{H \times W \times 2} \) in \( F_2 \) given the current depth value:

\[ \mathcal{H}(\tilde{G}_{2D}) = K_2 E_2 E_1^{-1} d_i K_1^{-1} \mathcal{H}(G_{2D}) \in \mathbb{R}^{H \times W \times 3}, \]  

(9)

where \( \mathcal{H}(G_{2D}) \in \mathbb{R}^{H \times W \times 3} \) denotes the homogeneous coordinates of the grid coordinates \( G_{2D} \in \mathbb{R}^{H \times W \times 2} \). Next, we perform bilinear sampling on \( F_2 \) with \( \tilde{G}_{2D} \) and obtain \( F_i^0 \in \mathbb{R}^{H \times W \times D} \) for depth candidate \( d_i \). Their correlation is then computed as

\[ C^i = \frac{F_1 \cdot F_i^0}{\sqrt{D}} \in \mathbb{R}^{H \times W}, \quad i = 1, 2, \ldots, N, \]  

(10)

where \( \cdot \) is the dot-product operation on the feature dimension \( D \). Concatenating the correlations for all depth candidates we obtain

\[ C_{\text{depth}} = [C^1, C^2, \ldots, C^N] \in \mathbb{R}^{H \times W \times N}. \]  

(11)

Similar to flow and stereo, we normalize the last dimension of \( C_{\text{depth}} \) and obtain the matching distribution

\[ M_{\text{depth}} = \text{softmax}(C_{\text{depth}}) \in \mathbb{R}^{H \times W \times N}. \]  

(12)

Finally, the depth is estimated by computing the weighted average of the matching distribution \( M_{\text{depth}} \) with all the depth candidates

\[ G_{\text{depth}} = [d_1, d_2, \ldots, d_N] \in \mathbb{R}^N; \]

\[ V_{\text{depth}} = M_{\text{depth}} G_{\text{depth}} \in \mathbb{R}^{H \times W}. \]  

(13)

Thus far, we have presented the detailed matching layers for all three tasks. We remark that all matching layers are differentiable and parameter-free, which not only enables end-to-end training but also doesn’t introduce any task-specific learnable parameters. We name our models for flow, stereo and depth tasks GMFlow, GMStereo and GMDepth, respectively, which represent our unified Global Matching formulation. Next, we will discuss our model for extracting strong features from the input images.

B. Feature Extraction

Key to our formulation lies in obtaining high-quality discriminative features for matching. To achieve this, we combine a common convolutional network (CNN) with a Transformer [27] as the feature extractor. More specifically, we first use a weight-sharing ResNet [71] to extract \( 8 \times \) downsampled features to keep computation tractable, similar to previous flow methods [17], [21]. However, the two features from the CNN are extracted independently, without considering their mutual relations yet. Integrating knowledge from the potential matching candidates in another image can intuitively enhance the feature’s distinctiveness and surpass ambiguities, as demonstrated by sparse matching methods [84]. This can be naturally implemented with the cross-attention mechanism, which is able to selectively aggregate information from another image by measuring cross-view feature similarities. We also use a self-attention layer to further improve the feature’s quality by considering larger context than the convolutional layer, and a two-layer feed-forward network (FFN, i.e., MLP) to further increase the capacity of the network following the original Transformer [27]’s design. The self-attention, cross-attention and FFN constitute a Transformer block, and our final Transformer architecture is a stack of six Transformer blocks which gradually improve the performance (Table I).

Specifically, for the extracted convolutional features \( \hat{F}_1 \) and \( \hat{F}_2 \), we first add fixed 2D sine and cosine positional encodings (following DETR [85]) to the features since they lack spatial information. Adding the position information also makes the matching process consider not only the feature similarity but also their spatial distance, which can help resolve ambiguities and improve performance (Table II(a)). Then the features are fed into the Transformer for feature enhancement. More specifically, for self-attention, the query, key and value in the attention mechanism [27] are the same feature. For cross-attention, the key and value are same but different from the query to model cross-view interactions. This process is performed for both \( \hat{F}_1 \) and \( \hat{F}_2 \) symmetrically:

\[ F_1 = \mathcal{T}(\hat{F}_1 + P, \hat{F}_2 + P), \quad F_2 = \mathcal{T}(\hat{F}_2 + P, \hat{F}_1 + P), \]  

(14)

where \( \mathcal{T} \) is a Transformer, \( P \) is the positional encoding, the first input of \( \mathcal{T} \) is query and the second is key and value.

One issue with the standard Transformer architecture [27] is the quadratic computational complexity due to the pair-wise attention operation. To improve efficiency, we adopt the shifted local window attention strategy from Swin Transformer [86]. However, unlike Swin that uses a fixed window size, we split the
feature to fixed number of local windows to make the window size adaptive with the feature’s spatial size. In this way, the attention mechanism can model long-range dependencies on high-resolution feature maps and accordingly better performance for large displacements can be achieved. Specifically, we split the input feature of size $H \times W$ to $K \times K$ windows (each with size $\frac{H}{K} \times \frac{W}{K}$), better for large displacements if $K$ is smaller, see Table II(b), and perform self- and cross-attentions within each local window independently. For every two consecutive local windows, we shift the window partition by $\left( \frac{H}{K} \times \frac{W}{K} \right)$ to introduce cross-window connections. In our method, we split into $2 \times 2$ windows (each with size $\frac{H}{2} \times \frac{W}{2}$), which leads to a good speed-accuracy trade-off (Table II(b)).

We note that for the rectified stereo matching task, since the correspondences lie only on the 1D horizontal direction, it’s redundant to perform full 2D cross-attention in the Transformer. Thus we perform 1D horizontal cross-attention for the stereo matching task, which is not only faster, but also leads to better performance (Table VI). Since only the linear feature projection layers of the Transformer are learnable, the final models are not affected by the specific parameter-free attention operation (2D, 1D or any other forms). Thus all the learnable parameters remain exactly the same for all three tasks.

C. Propagation

Our matching-based formulation implicitly assumes that corresponding pixels are visible in both images and thus they can be matched by comparing their similarities. However, this assumption will be invalid for occluded and out-of-boundary pixels, producing unreliable results in these regions (Fig. 3). To remedy this, by observing that the flow/disparity/depth field and the image itself share high structure similarity [30], [31], we propose to propagate the high-quality flow/disparity/depth predictions to unmatched regions by measuring feature self-similarity. This operation can be implemented efficiently with a single self-attention layer (illustrated in Fig. 2):

$$\hat{V} = \text{softmax} \left( \frac{F \hat{F}^T}{\sqrt{D}} \right) V \in \mathbb{R}^{H \times W \times 2},$$  \hspace{1cm} (15)

where $V$ is the flow/disparity/depth prediction from the softmax matching layer in Section III-A. Note that we don’t explicitly differentiate matched and unmatched pixels, but simply learn such a propagation process with ground truth flow supervision. Fig. 3 shows that this strategy can effectively correct the errors in unmatched regions.

Our current estimate is at the $8 \times 8$ downsampled feature resolution. To get the original image resolution prediction, we use RAFT’s upsampling [21] method that computes the full resolution flow/disparity/depth at each pixel as a weighted combination of a $3 \times 3$ grid of its coarse resolution neighbors. The combination weights are learned with a small 2-layer convolutional network, whose output channel is $8 \times 8 \times 3 \times 3$ for $8 \times 8$ upsampling. Fig. 2 provides an overview of our unified model.

D. Refinement

Our method presented so far (based on $1/8$ features) already achieves competitive performance while being simple and efficient. It can be further improved by using additional refinement steps, yielding different speed-accuracy trade-offs. We explore two types of refinement in this paper: hierarchical matching refinement with higher-resolution ($1/4$) features and local regression refinement with convolutions. We remark that the hierarchical matching refinement uses our matching-based formulation and thus is task-agnostic, while the local regression refinement is task-specific but optional. It can hence be viewed as a post-processing step to further improve the performance of our unified method.

1) Hierarchical Matching Refinement: Our unified global matching is performed at $1/8$ feature resolution, and a $1/8$ flow/disparity/depth prediction is obtained. Using additional higher-resolution ($1/4$) features for matching can further improve the performance and fine-grained details, while not introducing any task-specific learnable parameters as it uses our matching-based formulation. However, we found the improvement for unrectified stereo depth estimation to be not as significant as flow and stereo, and thus we choose to not perform hierarchical matching at $1/4$ resolution for the depth task. Specifically, for optical flow and rectified stereo matching tasks, we first upsample the $1/8$ flow/disparity prediction to $1/4$ resolution, and then warp the second CNN feature with the upsampled flow/disparity. In this way, the remaining task is reduced to matching between the original first CNN feature and the warped second CNN feature, and thus the same model depicted in Fig. 2 can be used at $1/4$ resolution but in a local range for refinement. More specifically, we perform a $9 \times 9$ local window matching for optical flow, and 1D horizontal local matching with length 9 for stereo matching (similar formulations as Section III-A1 and Section III-A2 but in a local range). The predicted flow/disparity residual is then added to the previous upsampled flow/disparity prediction obtained by global matching. For the Transformer, we split the $1/4$ feature map to $8 \times 8$ local windows (each with $1/32$ of the original image resolution) in attention computation to model local-range interactions. Next, we perform a $3 \times 3$ local window self-attention operation for flow/disparity propagation (similar formulation as Section III-C but in a local range). Finally, the $1/4$ flow/disparity prediction is obtained and it’s upsampled to the full resolution.

We note that we share the Transformer and self-attention weights in the $1/8$ and $1/4$ hierarchical matching stages since they perform basically very similar matching process except for different ranges (global vs. local). This not only reduces parameters but also improves generalization, as shown in the original GMFlow [38] paper. To generate the $1/4$ and $1/8$ resolution features, we take a similar approach to TridentNet [87]. Specifically, we first obtain a $1/4$ resolution feature map with
task-specific: for optical flow, we use 2D correlation; for rectified stereo matching, we also use 2D correlation since we found it to perform better than 1D correlation (Table VI) although some redundancy exists; for unrectified stereo depth, we use 2D correlation constructed from the current depth prediction and relative camera transformation. Such refinement architectures are task-specific and not shared across tasks. The optional number of additional iterative refinements is also different for different tasks, we choose this number empirically. More specifically, for optical flow, we use 6 additional refinement steps at 1/4 feature resolution after the hierarchical matching refinement; for rectified stereo matching, we use 3 additional refinement steps at 1/4 feature resolution after the hierarchical matching refinement; for unrectified stereo depth estimation, we use 1 additional refinement step at 1/8 feature resolution and no hierarchical matching is used. Note that the number of refinement steps as part of our post-processing is much less than previous pure iterative architectures (e.g., 31 refinements in RAFT [21] and its recent variants [31, 50, 51]) thanks to our stronger base model.

E. Training Loss

We supervise all predictions (including the intermediate network outputs and final ones) with the ground truth:

$$L = \sum_{i=1}^{N} \gamma^{N-i} \ell(V_i, V_g),$$  \hspace{1cm} (16)

where $N$ is the total number of predictions, and $\gamma$ (set to 0.9) is the weight that is exponentially increasing to give higher weights for later predictions following RAFT [21].

The definition of the loss function $\ell$ are following previous methods. More specifically, for optical flow, we use an $L_1$ loss [21]; for rectified stereo matching, we use the smooth $L_1$ loss [19]; for unrectified stereo depth estimation, we use the $L_1$ loss on the inverse depth [37]. Following [73], we also use an additional gradient loss for unrectified stereo depth:

$$L_{\text{grad}} = \sum_{i=1}^{N} \gamma^{N-i} (\ell(\partial_x V_i, \partial_x V_g) + \ell(\partial_y V_i, \partial_y V_g)), \hspace{1cm} (17)$$

where $\ell$ is the $L_1$ loss. The total loss for the depth task is a combination of the inverse depth loss and the gradient loss, where the combination weights are both 20.

IV. EXPERIMENTS

In this section, we will first study the properties of our unified model for each task independently, and then show the unique advantage of our unified model by cross-task transfer, and finally perform system-level comparisons with previous methods on standard benchmarks. The implementation details are presented in the supplementary material, available online.

A. Optical Flow

Datasets and evaluation setup: Following previous optical flow methods [16, 17, 21], we first train on the FlyingChairs (Chairs) [11] and FlyingThings3D (Things) [58] datasets, and then evaluate on Sintel [13] and KITTI [14] training sets for cross-dataset generalization. We also evaluate on the Things
validation set to see how the model performs on the same-domain data. Finally, we perform additional fine-tuning on Sintel and KITTI training sets and report the performance on the online benchmarks.

Metrics: We adopt the commonly used metric in optical flow, i.e., the end-point-error (EPE), which is the average $\ell_2$ distance between the prediction and ground truth. For the KITTI dataset, we also use $F1$-all, which reflects the percentage of outliers. To better understand the performance gains, we also report the EPE for different motion magnitudes. Specifically, we use $s_{10}$, $s_{10}$ to denote the EPE over pixels with ground truth flow motion magnitude falling into the ranges of $0 - 10$, $10 - 40$ and more than 40 pixels, respectively.

1) Methodology Comparison. Flow estimation approach: We compare our Transformer and softmax-based flow estimation method with cost volume and convolution-based approaches. Specifically, we adopt the state-of-the-art cost volume construction method in RAFT [21] that concatenates 4 local cost volumes at 4 scales, where each cost volume has a dimension of $H \times W \times (2R + 1)^2$. Here $H$ and $W$ denote the feature’s spatial size, and the search range $R$ is set to 4 following RAFT. To regress flow, we stack different numbers of convolutional residual blocks [71] to see how the performance varies. The final optical flow is obtained with a $3 \times 3$ convolution with 2 output channels. For our method, we stack different numbers of Transformer blocks for feature enhancement and the final optical flow is obtained with a global correlation and softmax layer. Table I shows that the performance improvement of our method is more significant compared to the cost volume and convolution-based approach.

For instance, our method with 2 Transformer blocks is already able to outperform 8 convolution layers, while a larger size ($s_{40+}$). The performance can be further improved by stacking more layers, surpassing the cost volume and convolution-based approach by a large margin. We also replace the Transformer in our model with a convolutional network for feature enhancement, which leads to a large drop in performance. This is largely due to the unique advantage of the cross-attention mechanism for modeling cross-view interactions (see Table II(a) for detailed evaluations of the Transformer components), which enables aggregation of the information from the other frame by considering cross-view similarities and thus greatly improves the quality of the extracted features. This is not achievable with convolutions [57].

Bidirectional Flow Prediction: Our method also simplifies backward optical flow computation by directly transposing the global correlation matrix in (1). Note that during training we only predict unidirectional flow while at inference, we can obtain bidirectional flow for free, without requiring to forward the network twice, unlike previous regression-based methods [44], [89]. The bidirectional flow can be used for occlusion detection with forward-backward consistency check (following [89]), as shown in Fig. 4.

2) Ablations. Transformer components: We ablate different Transformer components in Table II(a). The cross-attention contributes most, since it models the cross-view interactions between two features, which integrates the knowledge from another image and greatly improves the quality of the extracted features. Also, the position information makes the matching process position-dependent, which can help alleviate the ambiguities in pure feature similarity-based matching. Removing the feed-forward network (FFN) reduces a large number of parameters, while also leading to a moderate performance drop. The self-attention aggregates contextual cues within the same feature, leading to additional gains.

Local Window Attention: We compare the speed-accuracy trade-off of splitting the features into different numbers of local windows for attention computation in Table II(b). Recall that the extracted features from our CNN backbone have a resolution of 1/8, further splitting into $H/2 \times W/2$ local windows (i.e., 1/16 of the original image resolution) leads to a good trade-off between accuracy and speed, and thus is used in our model.

Matching Space: We replace our global matching (i.e., all-pair-wise matching $H \times W \times H \times W$ in Eq. (1)) with local matching (i.e., reduce the global matching in Eq. (1) to a local one $H \times W \times K \times K$ with window size $K \times K$) in Table II(c) and observe a significant performance drop, especially for large motion ($s_{40+}$). Besides, global matching can be computed efficiently with a simple matrix multiplication, while a larger size for local matching will be slower due to the excessive sampling operation.

Flow Propagation: Our flow propagation strategy results in significant performance gains in unmatched regions (including occluded and out-of-boundary pixels), as shown in Table II(d) and Fig. 3. The structural correlation between the feature and flow provides a valuable cue to improve the performance of pixels that are challenging to match.

3) Comparison With RAFT. Sintel: Table III shows the results on Things validation set and Sintel (clean and final) training sets after training on Chairs and Things training sets. Without using any refinement, our method achieves better performance on Things and Sintel (clean) than RAFT with 11 refinements. By using an additional task-agnostic hierarchical matching refinement at 1/4 feature resolution (Section III-D1), our method outperforms RAFT with 31 refinements, especially on large motion ($s_{40+}$). Fig. 5 visualizes the results. Furthermore, our model enjoys faster inference speed compared to RAFT and also does not require a large number of sequential processing. On the high-end A100 GPU, our model gains more speedup compared to RAFT’s sequential architecture (2.29× vs. 1.87×, i.e., ours: 151 → 66, RAFT: 170 → 91), reflecting that our method can benefit more from advanced hardware acceleration and demonstrating its potential for further speed optimization.

KITTI: Table V shows the generalization results on KITTI training set after training on Chairs and Things training sets. In this evaluation setting, our method doesn’t outperform RAFT, which is mainly caused by the gap between the synthetic training sets and the real-world testing dataset. One key reason behind our inferior performance is that RAFT, relying on fully convolutional neural networks, benefits from the inductive biases in convolution layers, which requires a relatively smaller size training data to generalize to a new dataset in comparison with Transformers [90], [91], [92], [93]. To substantiate this claim, we finetune both RAFT and our GMFlow on the additional Virtual KITTI 2 [94] dataset. The results in Table V verify that the performance gap becomes smaller when more data is available. We also train another version GMFlow+ that uses 6 additional local regression refinements (Section III-D2), we can observe
from Table V that GMFlow+ outperforms RAFT on KITTI dataset.

B. Stereo Matching

Datasets and Evaluation Setup: We first train on the synthetic Scene Flow [58] training set, and then evaluate on the Scene Flow test set and the KITTI 2015 [14] training set. Unlike previous representative stereo networks [19], [24], [25] that usually rely on a predefined disparity range (typically 192 pixels) to construct the local cost volume, our method is more flexible and can support unconstrained disparity prediction. To avoid extremely large disparity values in the data, we mask the pixels whose disparities exceed 400 pixels during both training and evaluation. Finally, we perform finetuning on KITTI 2015 Stereo, Middlebury Stereo, Argoverse Stereo and ETH3D Stereo datasets and report the performance on the online benchmarks.

Metrics: We adopt the commonly used metrics end-point-error (EPE) and D1-all, where EPE is the average ℓ1 distance between the prediction and ground truth disparity, and D1-all denotes the percentage of outliers.

1) Ablations: Stereo Cross-Attention: 1D vs. 2D: Unlike 2D optical flow, rectified stereo matching is a 1D correspondence task that corresponding pixels lie on the same horizontal scanline. Thus, it’s not necessary to perform 2D cross-attention in the Transformer to model cross-view interactions and 1D horizontal cross-attention is sufficient. As shown in Table VI, using 1D cross-attention is not only more efficient in terms of inference time (measured for KITTI resolution (384 × 1248) on a single V100 GPU), but also leads to better performance since unnecessary matching information is avoided. We note that the parameter-free cross-attention operation (2D, 1D or any other forms) doesn’t affect the learnable parameters (i.e., the linear projection layers) of the Transformer, and thus the pretrained model for optical flow and stereo matching tasks can still be shared.

Model Components: We ablate different components of our full model in Table IV. The results are consistent with those for the optical flow task in Tables II(a) and II(d). That is, the cross-attention contributes most, but the other components also contribute to the performance gains.

2) Comparison With RAFT-Stereo: We compare our GM-Stereo model with RAFT-Stereo [26] on the Scene Flow test set in Table VII. The prediction error and inference time for different number of refinement steps are reported. We can observe that our GM-Stereo model trained with random initialization (random init) already significantly outperforms RAFT-Stereo. Leveraging the pretrained GMFlow model as initialization (flow init) makes the performance gap even larger.

(a) Rectified stereo matching task

Cross-attention contributes most, consistent with the analysis in optical flow task (Table IIa).

(b) Unrectified stereo depth estimation task.
can further benefit from the pretrained flow model thanks to our unified model. As shown in Table VII, our GMStereo model trained with GMFlow model as initialization (flow init) leads to further performance boost, outperforming RAFT-Stereo by even larger margins.

C. Depth Prediction

Datasets and Evaluation Setup: For ablations, we train on the ScanNet [34] dataset, where we follow BA-Net [20] for the training and testing splits. Finally, we train and evaluate on the SUN3D [35], RGBD-SLAM [36] and Scenes11 [37] datasets for comparison with previous methods.

Metrics: Following previous methods [20], [72], we use 4 error metrics for evaluation of the depth quality, including Absolute Relative difference (Abs Rel), Squared Relative difference (Sq Rel), Root Mean Squared Error (RMSE) and RMSE in log scale (RMSE log).

1) Ablations. Model Components: We ablate different components of our full model in Table IV(a). The results are consistent with those for optical flow and stereo matching tasks in Tables II(a), II(b), and IV(a). That is, the cross-attention contributes most, and other components also contribute to the performance gains.

2) Comparison With Depth Field Network: The Depth Field Network (DeFiNe) [77] proposes an implicit way for learning cross-view correspondences, where the geometric priors (e.g., camera information) are encoded as inputs to a Transformer model for depth estimation. Different from DeFiNe, we learn task-agnostic features and obtain the depth prediction with a parameter-free matching layer. In Table VIII, we show a comparison with DeFiNe on ScanNet test set. Our GMDepth model achieves similar performance but has 4× less parameters and is 2× faster. It is also worth noting that DeFiNe relies on a series of 3D geometric augmentations to achieve competitive performance, while our GMDepth can be trained well without any such augmentations. Our method also has 4× less parameters and is 2× faster.

In Table VIII, we show a comparison with DeFiNe on ScanNet test set. Our GMDepth model achieves similar performance but has 4× less parameters and is 2× faster. It is also worth noting that DeFiNe relies on a series of 3D geometric augmentations to achieve competitive performance, while our GMDepth can be trained well without any such augmentations. Our method also has 4× less parameters and is 2× faster.

3) Comparison With DepthFormer: DepthFormer [74] proposes to use a Transformer to improve the quality of cost volume, while we leverage a Transformer to learn strong features for simple parameter-free matching. We compare with DepthFormer in Table IX by replacing our Transformer and matching layers with DepthFormer’s Transformer-enhanced cost volume and depth decoding layers. We train this model variant within our architecture and keep other components exactly the same. More specifically, we can directly use DepthFormer’s Transformer to learn strong features for the cost volume and further finetune the depth decoding layers. The visual results are shown in Fig. 6, where our model performs significantly better than a random initialized model. The pretrained flow model can be further finetuned for stereo and depth tasks, which not only leads to faster training speed, but also achieves better performance than random initialization (Table X).

D. Cross-Task Transfer

One unique benefit of our unified model is that it naturally enables cross-task transfer since all the learnable parameters are exactly the same. More specifically, we can directly use a pretrained optical flow model and apply it to both rectified stereo matching and unrectified stereo depth estimation tasks. As shown in Tables X(c) and X(d), our pretrained optical flow model performs significantly better than a random initialized model. The visual results are shown in Fig. 6, where our model achieves promising results. The pretrained flow model can be further finetuned for stereo and depth tasks, which not only leads to faster training speed, but also achieves better performance than random initialization (Table X).

We also experiment with transferring the pretrained models from the stereo and depth tasks to the optical flow task, but no obvious performance gain is observed. This is understandable since stereo and depth are both 1D correspondence problems, and their pretrained models might be specialized to the 1D correspondence matching task and thus are not able to bring clear benefits to the more general 2D correspondence task (i.e., optical flow).
13952 IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 45, NO. 11, NOVEMBER 2023

TABLE X
CROSS-TASK TRANSFER

(a) Flow to stereo transfer: error curves of disparity prediction error vs. numbers of training steps.

| Model              | Things EPE | Things D1 | KITTI EPE | KITTI D1 |
|--------------------|------------|-----------|-----------|----------|
| rand init, w/o ft | 75.24      | 96.06     | 103.47    | 96.95    |
| flow init, w/o ft | 2.56       | 18.19     | 1.98      | 17.60    |
| rand init, ft (50K) | 1.22      | 3.70      | 1.61      | 10.53    |
| flow init, ft (50K)| 1.10       | 3.04      | 1.39      | 7.56     |
| rand init, ft (100K)| 1.11       | 3.05      | 1.58      | 9.93     |
| flow init, ft (100K)| 1.00       | 2.77      | 1.37      | 7.38     |

(b) Flow to depth transfer: error curves of depth prediction error vs. numbers of training steps.

| Model              | Things Abs Rel | Things Sq Rel | Things RMSE | Things RMSE log |
|--------------------|----------------|---------------|--------------|-----------------|
| rand init, w/o ft | 0.536          | 1.309         | 1.300        | 0.584           |
| flow init, w/o ft | 0.198          | 0.364         | 0.599        | 0.235           |
| rand init, ft (50K)| 0.074          | 0.028         | 0.225        | 0.103           |
| flow init, ft (50K)| 0.066          | 0.023         | 0.203        | 0.092           |
| rand init, ft (100K)| 0.069         | 0.025         | 0.211        | 0.097           |
| flow init, ft (100K)| 0.063          | 0.021         | 0.193        | 0.088           |

(3) Flow to stereo transfer: performance comparison.

We show the comparisons of error curves between random initialization and using a pretrained optical flow model as initialization in Fig. 10a and Fig. 10b. The performance comparisons of different models without any finetuning or finetuned with different initialization (rand init vs. flow init) and different numbers of total training steps (50K vs. 100K) are shown in Table 10c and Table 10d.

TABLE XI
FLOW TO STEREO TRANSFER COMPARISON WITH RAFT

| Model                | #refine | EPE  | D1  | Time (ms) |
|----------------------|---------|------|-----|-----------|
| RAFT [21] (disparity from x-flow) | 0       | 8.10 | 34.12 | 15        |
|                      | 3       | 2.76 | 8.07 | 22        |
|                      | 7       | 2.08 | 6.33 | 31        |
|                      | 11      | 1.95 | 6.03 | 41        |
|                      | 31      | 1.93 | 5.90 | 95        |
| GMFlow (1D cross-attention, 1D matching) | 0       | 2.58 | 18.2 | 23        |
|                      | 1       | 1.38 | 5.27 | 58        |
| GMStereo, finetune (1D cross-attention, 1D matching) | 0       | 1.00 | 2.77 | 23        |
|                      | 1       | 0.89 | 2.64 | 58        |

The evaluations are conducted on Scene Flow test set for stereo matching task. For RAFT, we obtain the disparity from the x component of its 2D optical flow prediction. Our results are obtained by modifying the cross-attention function and the matching layer.

1) Flow to Stereo Transfer Comparison With RAFT: We compare with RAFT in terms of flow to stereo transfer in Table XI. More specifically, we use RAFT to extract optical flow from a stereo pair and obtain the disparity from the horizontal component of the 2D optical flow. For our method, we are able to obtain the disparity from our flow model GMFlow by modifying the parameter-free cross-attention function and the matching layer. We can observe from Table XI that our GMFlow with only 1 refinement already outperforms RAFT with 31 refinements, without any finetuning for the stereo task. Our unified model is able to benefit from additional finetuning and achieves further performance improvement.

2) Flow to Depth Transfer Comparison With RAFT: We compare with RAFT in terms of flow to depth transfer in Table XII. More specifically, we use RAFT to extract optical
flow from two posed images and obtain the depth prediction with triangulation [99]. For our method, we obtain the depth prediction by modifying the matching layer to the depth task. We can observe from Table XII that our method performs better than RAFT in terms of the ‘RMSE log’ metric, but inferior for other metrics. This indicates that our results might have large outliers that dominate the averaged metrics of ‘Abs Rel’, ‘Sq Rel’ and ‘RMSE’, but their influence becomes weaker when evaluated in the log scale. The possible reason for this phenomenon is that unlike stereo disparity that is a special case of optical flow, the depth matching layer is slightly different from the flow one, which might make it challenging to do direct cross-task transfer and some outliers might exceed. In contrast, the triangulation process involves solving a least-square problem, which is more complex than our simple argmax operation. However, one unique strength of our unified model is that the pretrained flow model can be finetuned for the depth task, and it can quickly adapt to the depth task and finally outperforms the triangulation-based approach by a large margin.

E. Benchmark Results

In this section, we perform system-level comparisons with previous methods on standard optical flow, stereo matching and depth estimation benchmarks.

1) Optical Flow: In Section IV-A3, we have demonstrated that our unified model with 1 additional task-agnostic hierarchical matching refinement at 1/4 feature resolution can already outperform 31-refinement RAFT. To fully unleash the potential of our method, we use additional task-specific post-processing steps for further improvement. More specifically, we use 6 additional RAFT’s iterative local regression refinements at 1/4 feature resolution, which can further improve our performance on unmatched regions and fine-grained details, as shown in Table XIII. We note that other post-processing strategies might also be applicable to our method, in this paper we adopt RAFT’s approach for convenience.

Sintel: The results on Sintel test set are shown in Table XIV. We achieve state-of-the-art results on the highly competitive Sintel (clean) dataset. On Sintel (final) dataset, our performance is only second to the recent FlowFormer [51] model, which uses a Transformer model that is pretrained on the large scale ImageNet dataset and is more computationally expensive due to the large number of sequential refinements like RAFT. The visual comparisons with RAFT are shown in Fig. 7. Our method can better capture the motion of fast-moving objects like the moving hand.

KITTI: The results are shown in Table XVII. We achieve competitive performance compared with the state-of-the-art methods LEAStereo [100] and CREStereo [62]. Besides, our model runs about 2× faster since we don’t rely on any 3D convolutions...
TABLE XV

| Method                  | Non-occluded pixels | All pixels |
|-------------------------|---------------------|------------|
| FlowNet2 [16]           | 6.94                | 10.41      |
| PWC-Net+ [95]           | 4.91                | 7.72       |
| HD3 [18]                | 3.93                | 6.55       |
| VCN [96]                | 3.89                | 6.30       |
| RAFT [21]               | 3.07                | 5.10       |
| CRAFT [50]              | 3.02                | 4.79       |
| SeparableFlow [49]      | 2.78                | 4.53       |
| GMFlowNet [98]          | 2.75                | 4.79       |
| DEQ-Flow [47]           | 2.96                | 4.91       |
| AGFlow [53]             | 2.97                | 4.85       |
| KPA-Flow [48]           | 2.82                | 4.60       |
| FlowFormer [51]         | 2.59                | 4.68       |
| GMFlow+                 | 2.40                | 4.49       |

TABLE XVI

| setup          | #refine | Things | KITTI | KITTI |
|----------------|---------|--------|-------|-------|
| baseline       | 1       | 0.94   | 2.95  | 1.31  | 6.79  |
| 1D correlation | 2       | 0.84   | 2.46  | 1.27  | 6.22  |
| 2D correlation | 4       | 0.82   | 2.32  | 1.32  | 6.50  |

We observe that 2D correlation is better than 1D correlation in the local correlation and convolution-based regression method.

TABLE XVII

| Model           | D1-all (All) | D1-all (None) | Time (s) |
|-----------------|--------------|---------------|----------|
| LEASTereo [100] | 1.65         | 1.51          | 0.30     |
| CRESTereo [62]  | 1.69         | 1.54          | 0.30     |
| GANet-deep [28] | 1.81         | 1.63          | 1.80     |
| CFNet [60]      | 1.88         | 1.73          | 0.18     |
| AAPNet [19]     | 2.03         | 1.85          | 0.06     |
| PSMNet [24]     | 2.32         | 2.14          | 0.41     |
| GMStereo        | 1.77         | 1.61          | 0.17     |

TABLE XVIII

| Model          | bad 2.0 | bad 4.0 | AvgErr | RMS | Time (s) |
|----------------|---------|---------|--------|-----|----------|
| CRESTereo [62] | 3.71    | 2.04    | 1.15   | 7.70| 3.55 (F) |
| RAFT-Stereo [21] | 4.74    | 2.75    | 1.27   | 8.41| 11.6 (F) |
| LEASTereo [100] | 7.15    | 2.75    | 1.43   | 8.11| 2.90 (H) |
| HSMNet [101]   | 10.2    | 4.83    | 2.07   | 10.3| 0.51 (F) |
| CFNet [60]     | 10.1    | 6.49    | 3.49   | 15.4| 0.69 (H) |
| GMStereo       | 7.14    | 2.96    | 1.31   | 6.45| 0.73 (F) |

"F" and "H" denote the results are generated using the full and half resolution images, respectively.

Middlebury: The results are shown in Table XVIII. Our GM-Stereo achieves the first place in terms of the RMS (Root Mean Square) disparity error metric. Besides, our method shows much higher efficiency than CRESTereo [62] (5× faster) and RAFT-Stereo [26] (15× faster) on such a high-resolution dataset. We also show some visual comparisons in Fig. 8, our method produces sharper object structures than CRESTereo [62] and RAFT-Stereo [26].

ETH3D: The results are shown in Table XIX. We achieve the second place in terms of the ‘bad 1.0’ and ‘bad 2.0’ metrics and the first place in terms of the ‘bad 4.0’ metric.

Argoverse: We also participate in the Argoverse Stereo Challenge held in the context of the CVPR 2022 Autonomous Driving Workshop to further demonstrate the potential of our method. Since this competition requires algorithms to produce disparity predictions in 200 ms or less, we use global matching at 1/8 feature resolution and two additional local regression refinements also at 1/8 resolution. The inference time is about 190 ms for a stereo pair of 1024 × 1232 resolution (resized from the full resolution for prediction). The results are shown in Table XX, where our GMStereo achieves the first place and clearly outperforms all other submissions.

3) Depth Estimation: For unrectified two-view depth estimation, our final model doesn’t use the hierarchical matching refinement at 1/4 feature resolution since we don’t observe very (unlike LEASTereo) or a large number (20+) of sequential refinements (unlike CRESTereo). Compared with previous lightweight stereo model AANet [19], our method performs much better. Besides, our model can be implemented with pure PyTorch, without requiring to build additional CUDA ops like AANet, which demonstrates that our method achieves a better speed-accuracy trade-off and has more practical advantages.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
large performance gains. Instead, we use an additional task-specific local regression refinement at 1/8 feature resolution, which further improves the performance while maintaining fast inference speed.

*ScanNet:* The results are shown in Table XXI. We achieve performance comparable to the representative method DeepV2D [73] and outperform DeMoN [37] and BA-Net [20] by a large margin. Notably, our model runs 10× faster than BA-Net and 15× faster than DeepV2D, which both heavily rely on computationally expensive 3D convolutions. This demonstrates that our model has strong potential for real-world use cases.

*RGBD-SLAM, SUN3D, and Scenes11:* The evaluation results on respective RGBD-SLAM, SUN3D and Scenes11 test sets are shown in Table XXII. We outperform previous representative methods (e.g., DPSNet) by a large margin. Compared with IIB [29] which injects the geometric inductive bias directly to the input of the Transformer, our performance is similar but our model is more lightweight and faster, which demonstrates that a better modeling of the geometric inductive bias enables the problem to be solved more efficiently.

### V. LIMITATION AND DISCUSSION

Our work has several limitations. First, our method still has room for further improvement in unmatched regions. As shown in Table XIV, the performance in matched regions on the Sintel dataset is already very accurate (with an end-point-error of 0.34 pixels on the clean split and 1.10 pixels on the final split). However, the error in unmatched regions is considerably larger, which deserves further investigation in future. Second, we resort to RAFT’s iterative refinement architecture as a post-processing step to further improve our performance. We believe more lightweight and effective approach would be applicable which we consider as interesting future work. Third, our full model is still not able to achieve real-time inference speed. Further improvements are necessary to enable applications with real-time requirements (20 FPS or more). Finally, in this paper, we have demonstrated the applicability of our method to multiple 2-frame tasks. We consider the extension of our approach to the multi-view scenario as an interesting future direction.

Our unified model might also shed some light on training a single model to do all tasks simultaneously. In this paper, we haven’t shown such experiments yet. There are also additional challenges to resolve (e.g., how to balance different tasks in the joint training process). Besides, to train such a unified model, one could also explore recent unsupervised pretraining approaches (e.g., masked autoencoders [103]) to learn general feature representations for matching. We believe that our work may serve as a fruitful basis for further research in this area.

### VI. CONCLUSION

We have presented a unified formulation and model for three different motion and 3D perception tasks: optical flow, rectified stereo matching and unrectified stereo depth estimation. We demonstrated that all three tasks can be solved with a unified model by formulating them as a unified dense correspondence matching problem. This allows to reduce the problem to learning high-quality discriminative features for matching, for which we use a Transformer, in particular exploiting its cross-attention mechanism to integrate information from the other view. One unique benefit of our unified model is that it naturally enables cross-task transfer since all the learnable parameters are exactly the same. Our final model achieves state-of-the-art or highly competitive performance on 10 popular flow/stereo/depth datasets, while being simpler and more efficient in terms of model design and inference speed.

A key result of this paper is that features aggregated via a Transformer from both input images are stronger and contain more discriminative correspondence information, which enables to greatly simplify existing motion and depth estimation pipelines, while achieving improved performance. We hope our findings can be useful for more dense correspondence and multi-view perception tasks.

### REFERENCES

[1] R. Hartley and A. Zisserman, *Multiple view geometry in Comput. Vis.* Cambridge, U.K.: Cambridge Univ. Press, 2003.

[2] D. Sun, S. Roth, and M. J. Black, “Secrets of optical flow estimation and their principles,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2432–2439.

[3] S. Agarwal et al., “Building Rome in a day,” *Commun. ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[4] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[5] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *Proc. IEEE/ACM 6th Int. Symp. Mixed Augmented Reality*, 2007, pp. 225–234.

[6] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite;” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[7] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artif. Intell.*, vol. 17, no. 1/3, pp. 185–203, 1981.

**TABLE XXI**

| Model     | Abs Rel | Sq Rel | RMSE | RMSE log | Time (s) |
|-----------|---------|--------|------|----------|----------|
| DeMoN [37]| 0.231   | 0.520  | 0.761| 0.289    | 0.69     |
| BA-Net [20]| 0.161  | 0.092  | 0.346| 0.214    | 0.38     |
| DeepV2D [73]| 0.057  | 0.010  | 0.168| 0.080    | 0.69     |
| GMDepth   | 0.059   | 0.019  | 0.179| 0.062    | 0.04     |

**TABLE XXII**

| Dataset | Model     | Abs Rel | Sq Rel | RMSE | RMSE log |
|---------|-----------|---------|--------|------|----------|
| **RGBD-SLAM** | DeMoN [37] | 0.157   | 0.497  | 0.182| 0.157    |
|          | DeepMVS [102]| 0.294  | 0.430  | 0.868| 0.351    |
|          | DPSNet [72]  | 0.154   | 0.215  | 0.723| 0.226    |
|          | IIB [29]     | 0.095   | 0.350  | 0.560| -        |
|          | GMDepth      | 0.101   | 0.177  | 0.556| 0.167    |
| **SUN3D** | DeMoN [37] | 0.214   | 1.120  | 2.421| 0.206    |
|          | DeepMVS [102]| 0.282  | 0.435  | 0.944| 0.363    |
|          | DPSNet [72]  | 0.147   | 0.107  | 0.427| 0.191    |
|          | IIB [29]     | 0.099   | -      | 0.293| -        |
|          | GMDepth      | 0.112   | 0.068  | 0.336| 0.146    |
| **Scenes11**| DeMoN [37] | 0.556   | 3.402  | 2.603| 0.391    |
|          | DeepMVS [102]| 0.210  | 0.373  | 0.891| 0.270    |
|          | DPSNet [72]  | 0.056   | 0.144  | 0.714| 0.140    |
|          | IIB [29]     | 0.056   | -      | 0.523| -        |
|          | GMDepth      | 0.050   | 0.069  | 0.491| 0.106    |
[25] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, “GA-Net: Guided

[24] J.-R. Chang and Y.-S. Chen, “Pyramid stereo matching network,” in

[23] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-

[22] G. Xu, J. Cheng, P. Guo, and X. Yang, “Attention concatenation volume

[17] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “PWC-Net: CNNs for optical

[16] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A.Dosovitskiy, and T. Brox,

[14] M. Menze and A. Geiger, “Object scene flow for autonomous ve-

[13] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, “A naturalistic open

[12] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, “MVSNet: Depth inference

[10] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon,

[8] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual

[7] A. Zhang, Y. Song, Z. Liu, R. Wang, and D. Xue, “ScanNet: Richly-annotated 3D reconstructions of indoor scenes,” in

[6] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and

[5] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A bench-

[4] T. Brox and J. Malik, “Large displacement optical flow: Descriptor

[3] S. Bai, Z. Geng, Y. Savani, and J. Z. Koller, “Deep equilibrium optical

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, “High accuracy optical

[1] J. Xiao, A. Owens, and A. Torralba, “SUN3D: A database of big spaces

[58] N. Mayer et al., “A large dataset to train convolutional networks for

[56] K.-J. Yoon and I. S. Kweon, “Adaptive support-weight approach for

[55] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, “Fast cost-

[54] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, “EpicFlow:

[53] A. Luo, F. Yang, X. Li, and S. Liu, “Learning optical flow with kernel

[52] Z. Zheng et al., “DIP: Deep inverse patchmatch for high-resolution

[51] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-

[50] M.-F. Chang et al., “Argoverse: 3D tracking and forecasting with rich

[49] S. Bai, Z. Geng, Y. Savani, and J. Z. Koller, “Deep equilibrium optical

[48] S. Bai, Z. Geng, Y. Savani, and J. Z. Koller, “Deep equilibrium optical

[47] S. Bai, Z. Geng, Y. Savani, and J. Z. Koller, “Deep equilibrium optical

[46] A. Zhang, Y. Song, Z. Liu, R. Wang, and D. Xue, “ScanNet: Richly-annotated 3D reconstructions of indoor scenes,” in

[45] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A bench-

[44] T. Brox and J. Malik, “Large displacement optical flow: Descriptor

[43] A. Meyer, M. Menze, and A. Geiger, “A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation,” in

[42] W. Vifan, C. Doersch, R. Arandjelovic, J. Carreira, and A. Zisserman,

[41] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, “High accuracy optical

[40] T. Brox and J. Malik, “Large displacement optical flow: Descriptor

[39] A. Bruhn, J. Weickert, and C. Schnörr, “Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods,” in

[38] B. Ummenhofer et al., “DeMoN: Depth and motion network for learning

[37] B. Ummenhofer et al., “DeMoN: Depth and motion network for learning

[36] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A bench-

[35] J. Xiao, A. Owens, and A. Torralba, “SUN3D: A database of big spaces

[34] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and

[33] M.-F. Chang et al., “Argoverse: 3D tracking and forecasting with rich

[32] M.-F. Chang et al., “Argoverse: 3D tracking and forecasting with rich

[31] S. Jiang, D. Campbell, Y. Lu, H. Li, and R. Hartley, “Learning to estimate

[30] T.-W. Hui and C. C. Loy, “LiteFlowNet3: Resolving correspondence

[29] H. Xu, J. Yang, J. Cai, J. Zhang, and X. Tong, “High-resolution optical

[28] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “PWC-Net: CNNs for optical

[27] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-

[26] H. Xu and J. Zhang, “AA-Net: Adaptive aggregation network for efficient

[25] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A bench-

[24] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, “High accuracy optical

[23] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-

[22] A. Zhang, Y. Song, Z. Liu, R. Wang, and D. Xue, “ScanNet: Richly-annotated 3D reconstructions of indoor scenes,” in
Xu et al. “UNIFYING FLOW, STEREO AND DEPTH ESTIMATION”

[59] A. Kendall et al., “End-to-end learning of geometry and context for deep stereo regression,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 66–75.

[60] Z. Shen, Y. Dai, and Z. Rao, “CFNet: Cascade and fused cost volume for fast stereo matching,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 13 906–13 915.

[61] M. Poggi, F. Tosi, K. Batsos, P. Mordohai, and S. Mattoccia, “On the synergies between machine learning and binocular stereo for depth estimation from images: A survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 9, pp. 3514–3534, Sep. 2022.

[62] J. Li et al., “Practical stereo matching via cascaded recurrent network with adaptive correlation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 16 263–16 272.

[63] Z. Li et al., “Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 6197–6206.

[64] W. Guo et al., “Context-enhanced stereo transformer,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 263–279.

[65] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, “Unsupervised CNN for single view depth estimation: Geometry to the rescue,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 740–756.

[66] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 270–279.

[67] 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. 1851–1858.

[68] H. Xu, J. Zheng, J. Cai, and J. Zhang, “Region deform network for unsupervised depth estimation from unconstrained monocular videos,” in Proc. Int. Joint Conf. Artif. Intell., 2019.

[69] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 3828–3838.

[70] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 3, pp. 1623–1637, Mar. 2023.

[71] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[72] S. Im, H.-G. Jeon, S. Lin, and I. S. Kweon, “DPSNet: End-to-end deep plane sweep stereo,” in Proc. Int. Conf. Learn. Representations, 2019.

[73] Z. Teed and J. Deng, “Deep2v2d: Video to depth with differentiable structure from motion,” in Proc. Int. Conf. Learn. Representations, 2020.

[74] V. Guizilini, R. Ambru, D. Chen, S. Zakharov, and A. Gaidon, “Multi-frame self-supervised depth with transformers,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 160–170.

[75] S. S. Sayed, J. Gibson, C. Watson, V. Prisacariu, M. Firman, and C. Godard, “SimpleRecon: 3D reconstruction without 3D convolutions,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 1–19.

[76] Z. Ma, Z. Teed, and J. Deng, “Multiview stereo with cascaded epipolar raft,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 734–750.

[77] V. Guizilini et al., “Depth field networks for generalizable multi-view scene representation,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 245–262.

[78] J. Watson, O. Mac Aodha, V. Prisacariu, G. Brostow, and M. Firman, “The temporal opportunistic: Self-supervised multi-frame monocular depth,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 1164–1174.

[79] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan, “Cascade cost volume for high-resolution multi-view stereo and stereo matching,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 2495–2504.

[80] Y. Ding et al., “TransMVSNet: Global context-aware multi-view stereo network with transformers,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 8585–8594.

[81] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely, “Learning feature descriptors using camera pose supervision,” in Proc. Eur. Conf. Comput. Vis., 2021, pp. 757–771.

[82] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 4104–4113.

[83] R. T. Collins, “A space-sweep approach to true multi-image matching,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 1996, pp. 358–363.

[84] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “SuperGlue: Learning feature matching with graph neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 4938–4947.

[85] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 213–229.

[86] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 9992–10002.

[87] Y. Li, Y. Chen, N. Wang, and Z. Zhang, “Scale-aware Trident networks for object detection,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 6054–6063.

[88] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2117–2125.

[89] S. Meister, J. Hur, and S. Roth, “UnFlow: Unsupervised learning of optical flow with a bidirectional census loss,” in Proc. AAAI Conf. Artif. Intell., 2018.

[90] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” 2021.

Xiao Fei Xu received the master’s degree from the University of Science and Technology of China, in 2021. He is currently working toward the PhD degree with the ETH Zurich and University of Tubingen, supervised by Fisher Yu and Andreas Geiger. He studied with Nanyang Technological University, Singapore and Monash University (remotely), Australia, and interned with Microsoft Research Asia. His research interests include flow, stereo, depth, and 3D scene representation learning. He received the Outstanding Reviewer award in CVPR 2022.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Jing Zhang (Senior Member, IEEE) is currently a research fellow with the School of Computer Science, University of Sydney. He has published more than 60 papers in prestigious conferences and journals, such as CVPR, ICCV, ECCV, NeurIPS, ICLR, IEEE Transactions on Pattern Analysis and Machine Intelligence, and International Journal of Computer Vision. His research interests include computer vision and deep learning. He is also a senior program committee member of the AAAI Conference on Artificial Intelligence and the International Joint Conference on Artificial Intelligence. He serves as a regular reviewer for many prestigious journals and conferences.

Jianfei Cai (Fellow, IEEE) received the PhD degree from the University of Missouri-Columbia. He is currently a professor and serves as the head of the Data Science and AI Department, Faculty of IT, Monash University, Australia. Before that, he had served as Head of Visual and Interactive Computing Division and Head of Computer Communications Division in Nanyang Technological University (NTU). His major research interests include computer vision, multimedia, and visual computing. He has successfully trained more than 30 PhD students with three getting NTU SCSE Outstanding PhD thesis award. He is a co-recipient of paper awards in ACCV, ICCM, IEEE ICIP and MMSP. He serves or has served as an associate editor for International Journal of Computer Vision, IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, and IEEE Transactions on Circuits and Systems for Video Technology as well as serving as area chair for CVPR, ICCV, ECCV, IJCAI, ACM Multimedia, ICME, and ICIP. He was the chair of IEEE CAS VSPC-TC during 2016-2018. He has also served as the leading TPC chair for IEEE ICME 2012 and will be the leading general chair for ACM Multimedia 2024.

Hamid Rezatofighi received the PhD degree from the Australian National University, in 2015. He is a lecturer with the Faculty of Information Technology, Monash University, Australia. Previously, he was an endeavour research fellow with the Stanford Vision Lab (SVL), Stanford University, and a senior research fellow with the Australian Institute for Machine Learning (AIML), University of Adelaide. He has published more than 60 top tier papers in computer vision, AI and machine learning, robotics, and has been involved as primary investigator in several successful grant fundings (in total over 11.0 M), including two DARPA grants and one ARC discovery grant. He served as the publication chair in ACCV 2018 and as the area chair in WACV 2021, CVPR 2020, and CVPR 2022-2023. His research interests include computer vision, machine learning, and robotic vision and the visual perception problems that are required for an autonomous robot to navigate or interact in a human crowded environment.

Fisher Yu received the PhD degree from Princeton University. He is an assistant professor with ETH Zurich in Switzerland. He became a postdoctoral researcher with UC Berkeley afterwards. He directs the Visual Intelligence and Systems (VIS) Group. His research interests include the junction of machine learning, computer vision, and robotics.

Ducheng Tao (Fellow, IEEE) is currently a professor of computer science, Peter Nicol Russell Chair and an Australian laureate fellow with the Sydney AI Center and the School of Computer Science, Faculty of Engineering, University of Sydney. He mainly applies statistics and mathematics to artificial intelligence and data science, and his research is detailed in one monograph and more than 200 publications in prestigious journals and proceedings at leading conferences. He received the 2015 and 2020 Australian Eureka Prize, the 2018 IEEE ICDM Research Contributions Award, and the 2021 IEEE Computer Society McCluskey Technical Achievement Award. He is a fellow of the Australian Academy of Science, AAAS, and ACM.

Andreas Geiger received the PhD degree from the Karlsruhe Institute of Technology (KIT), in 2013. He is a professor with the University of Tübingen. Previously, he was a visiting professor with ETH Zürich and a group leader with the Max Planck Institute for Intelligent Systems. He studied with KIT, EPFL, and MIT. His research interests include the intersection of computer vision, machine learning and robotics, with a particular focus on 3D scene perception, deep representation learning, generative models, and sensori-motor control in the context of autonomous systems. He maintains the KITTI and KITTI-360 benchmarks.