IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 January 2024, accepted 10 March 2024, date of publication 22 March 2024, date of current version 1 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3380917

RESEARCH ARTICLE

Lightweight Optical Flow Estimation Using 1D Matching

WONYONG SEO[1], WOONSUNG PARK[2], AND MUNCHURL KIM[1], (Senior Member, IEEE)
[1]Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea
[2]Samsung Electronics, Suwon-si 16677, South Korea

Corresponding author: Munchurl Kim (mkimee@kaist.ac.kr)

This work was supported in part by Samsung Electronics, and in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by Korean Government [Ministry of Science and Information and Communications Technology (MSIT)] (Deep Learning Based Visual Representational Learning and Rendering of Static and Dynamic Scenes) under Grant RS-2022-00144444.

ABSTRACT Recent advancements in optical flow estimation have led to notable performance gains, driven by the adoption of transformer architectures, enhanced data augmentation, self-supervised learning techniques, the use of multiple video frames and iterative refinement of estimate optical flows. Nonetheless, these cutting-edge methods encounter substantial challenges with surge in computational complexity and memory demands. In response, we introduce a lightweight optical flow method, called MaxFlow, to address the trade-off between computational complexity and prediction performance. By leveraging MaxViT, we design a network with a global receptive field at reduced complexity, and proposed 1D matching to alleviate the computational complexity from $(H \times W)^2$ to $H \times W(H + W)$, wher H and W denotes height and width of input image. Consequently, our method achieves the lowest computational complexity compared to both state of the arts(SOTA) and other lightweight optical flow estimation methods, while still achieving competitive results with the SOTA techniques. We performed extensive experiments to show the effectiveness of our method, achieving about 5 to 6 times reductions in computation complexity while maintaining the prediction accuracy with only degradation of 16% in term of end point error(EPE) at Sintel test clean sequences with respect to RAFT method.

INDEX TERMS 1D matching, MaxViT, optical flow estimation, lightweight optical flow estimation, positional embedding, transformer.

I. INTRODUCTION

Optical flow prediction or estimation, one of the most classical problems in the field of computer vision, involves predicting the movement of pixels between two successive frames. This prediction is essential for various sub-tasks such as action recognition, video super-resolution, video frame interpolation, and autonomous driving etc.

With the advent of deep learning, traditional energy minimization-based optical flow estimation has been supplanted by learnable networks. Deep learning-based methods for optical flow estimation rely on the similarity information between two image frames, commonly referred to as the cost volume [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]. This approach involves identifying the corresponding pixels between two images to facilitate precise optical flow estimation. The early deep learning-based techniques for optical flow estimation, such as Flownet [1], PWCNet [3], and SpyNet [4], utilized local cost volumes, focusing on the relationship between each pixel in a preceding frame and its nearby pixels in a following frame, to conduct optical flow estimation.

In the evolving landscape of deep learning, the emergence of vision transformers has catalyzed notable advancements in diverse areas, including optical flow estimation. The state-of-the-art (SOTA) methods such as RAFT [5] and GMFlow [6] have achieved unprecedented accuracy, yet they grapple with increased computational demands and memory consumption.

The associate editor coordinating the review of this manuscript and approving it for publication was Ziyan Wu.

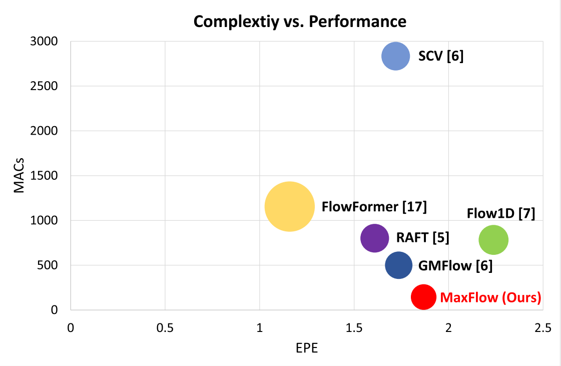

FIGURE 1. Comparison of model complexity and performance on Sintel datasets (consists of 448 × 1024 resolution images).between SOTA and lightweight optical flow estimation methods. The model complexity and prediction accuracy are measured in terms of MACs (multiplications and acculmulations calculations) and EPE (end point error). The circle size is proportional to the number of each model's parameters.

This is predominantly due to the computation of all-pair cost volumes, scaling quadratically with the $H \times W$ resolution of input images, denoted by $(HW)^2$.

The burgeoning demand for high-resolution video processing further exacerbates these resource constraints, making efficient optical flow estimation more challenging. One of the current efforts to mitigate the increase of computation complexity involves 1-dimensional and sparse cost volume approaches [7], [9]. However, these strategies still incur considerable computational overhead, largely attributed to their dependence on repetitive GRU [11], [12] architectures. A promising innovation, the global matching approach [6], circumvents the dependency on GRUs for accurate optical flow prediction. Nevertheless, this method incorporates a Swin Transformer [13] layer with a window size of $[H/2, W/2]$, where computations and memory usage continue to scale with $(HW)^2$. This limitation constrains its practical utility, especially in high-resolution contexts.

Our paper focuses on addressing the challenge of computation efficiency in optical flow estimation, while maintaining competitive performance. Fig. 1 shows the comparison of our lightweight optical flow estimation method (MaxFlow) with other state-of-the-art (SOTA) methods in terms of model complexity and prediction performance. The contributions of our work are summarized as follows:

- We introduce a novel lightweight optical flow estimation method that leverages a global matching scheme of low complexity in conjunction with the multi-axis vision transformer architecture [14];
- For this, the architecture of our method is designed to exhibit a global receptive field, with computational complexity proportional to HW. This significant reduction from the quadratic scaling of HW in the existing methods positions our approach as a viable solution for high-resolution video processing, marrying efficiency with high performance in optical flow estimation;
- Additionally, we propose an 1D matching method for correspondence finding, which allows to effectively perform optical flow estimation with a reduced computational complexity and memory footprint compared to conventional global matching methods.
- Finally, our proposed lightweight optical flow estimation method yields the best trade-off between the computation and prediction accuracy compared to the recent lightweight SOTA methods.

II. RELATED WORKS

A. OPTICAL FLOW ESTIMATION

Traditional approaches to optical flow estimation often minimized energy functions under brightness consistency and smoothness assumptions [15]. These assumptions are not always valid, as lighting conditions may differ between the input images, and the smoothness assumption may break down along motion boundaries.

The integration of deep learning into optical flow estimation marked a revolutionary shift in the field. The pioneering work, named FlowNet [1] introduced a deep learning architecture that was borrowed from semantic segmentation, notably an encoder-decoder structure, and applied it for optical flow estimation. Their framework utilized a cost volume, constructed through the dot product of features from consecutive frames, to quantify feature similarities. Subsequently, Flownet2 was introduced by stacking multiple FlowNets, as described in [2]. This architecture demonstrated performance comparable to traditional optical flow estimation methods in terms of prediction accuracy while being much faster than the traditional non-deep-learning-based methods. In spite of its fast run time on GPU, Flownet2 faced limitations due to its high computational demands.

Seeking the computation efficiency, Spynet [16] was proposed, leveraging a multi-scale pyramid structure and warping techniques to reduce the computational burden. Their method achieved competitive results with Flownet2, but with a fraction of the parameters and computational requirements, thus expanding the practical applicability of optical flow estimation in various scenarios.

PWC-Net [3] further pushed the boundaries of computation efficiency in optical flow estimation. This method, similar to Spynet, employed a pyramid-based structure but introduced a significant innovation by processing pyramid structures and warping in the feature domain rather than the image domain. PWC-Net starts at the coarsest scale, incrementally refining the flow estimation as it progresses to finer scales. This approach not only reduced computational overhead but also improved the prediction performance of optical flow estimations.

RAFT [5] identified critical limitations in the existing methods, particularly in PWC-Net's ability to accurately capture motion for small or fast-moving objects. These objects often went unnoticed at the lowest scales of the pyramid structure, and the limited receptive field at finer scales hindered accurate motion capture. RAFT addressed these issues by calculating all-pair cost volumes at a reduced scale and utilizing Conv-GRU for iterative refinement, leading to significant improvements in prediction accuracy at the cost of increasing the computation complexity.

GMFlow [6] then addressed the high computational demands of RAFT [5], focusing on the iterative nature of its optical flow estimation. Traditional models addressed this problem through regression techniques, utilizing either local cost volume with warping [3], or all-pair cost volume with look-up operations [5]. GMFlow innovatively reframed optical flow estimation as a *matching problem*, a concept derived from stereo matching or feature matching tasks.

B. LIGHTWEIGHT OPTICAL FLOW ESTIMATION

Although very recent work on deep learning based optical flow estimation, such as RAFT [5], GMA [8], FlowFormer [17], has greatly improved the prediction performance, the computation complexity has also been dramatically increased, thus making it difficult the possible real time applicability. For example, RAFT [5], while groundbreaking, necessitated substantial GPU memory to store its 4D all-pair cost volume, which is especially problematic for high-resolution image input. Additionally, the extensive usage of ConvGRU iterations contributed to its significant computational load. Therefore, the need for more lightweight and computationally efficient optical flow estimation methods became necessitated.

As an effort to reduce such a computation complexity, Flow1D [7] tried to solve these challenging issues by introducing a 1D separable cost volume, thus reducing the computation and memory requirements of the 4D all-pair cost volume used in RAFT. By calculating the cost volumes in both vertical and horizontal directions separately, Flow1D managed to stably compute optical flow in high-resolution images with a reduced computational footprint.

Similarly, SCV [9] proposed the use of a sparse cost volume, arguing that storing all-pair cost volumes is inefficient as most pairs contribute little to the final optical flow estimation. By focusing on the most correlated features, SCV aimed to minimize memory usage while maintaining estimation accuracy.

Despite these advances in reducing the computation complexity of cost volumes, both Flow1D and SCV still required significant computational resources for their iterative ConvGRU operations to achieve high accuracy in optical flow estimation. The challenge thus remained to develop methods that could balance computational efficiency with high accuracy in flow estimation.

C. VISION TRANSFORMER (VIT)

The inception of transformer architectures in natural language processing (NLP) [18] introduced a paradigm shift away from the localized bias of convolutional neural networks (CNNs) and the sequential bias of recurrent neural networks (RNNs), towards a non-biased architecture. Transformers, devoid of these biases, have shown a tendency to overfit with smaller datasets but excel with larger volumes of data, demonstrating substantial performance gains in the NLP domain. The unveiling of the Vision Transformer (ViT) [19] marked the beginning of the widespread adoption of transformer structures into computer vision fields, leading to significant enhancements in performance when used as backbone architectures for various low- and high-level computer vision tasks.

However, vanilla vision transformers [19] are not without their limitations, particularly in their operation at a fixed input scale, their intensive computation and data requirements. To address these challenges, several studies have proposed modifications to the basic ViT structure [4], [20], [21]. Among these, the Swin Transformer [13] has emerged as a notable innovation, introducing window self-attention as a novel solution to reducing the computation complexity for global self-attention operations. This approach substantially reduces computational load by localizing attention calculations within non-overlapping windows instead of the entire images, while concurrently improving performance by imparting an appropriate locality bias. The Swin Transformer further mitigates the limitations of a fixed receptive field by employing shifted window attention, which allows for self-attention across moved windows, thus ensuring effective information transfer.

Despite its efficiencies in computation and performance, the Swin Transformer somewhat offsets the inherent advantage of global information transmission found in the original ViT, necessitating either numerous Swin layers or larger window sizes to achieve large receptive fields. Addressing this, Tu et al. [14] proposed an enhanced model that leverages attention within a multi-axis global grid. Following window self-attention, their model, referred to as MaxViT, constructs a novel grid along an additional axis to aggregate global features within a single window for self-attention. This design not only maintains the computational efficiency proportional to $H \times W$ but also retains a global receptive field, thus facilitating efficient global information exchange. By balancing the benefits of a global receptive field with reduced computational demands, MaxViT presents itself as a viable new baseline for transformer-based architectures in the field of computer vision.

III. METHODS

We propose a lightweight optical flow estimation (OFE) method, called MaxFlow, aiming to reduce computational load and memory usage while maintaining competitive accuracy in optical flow estimation, compared to the recent SOTA OFE models [5], [8], [17]. Fig. 2 illustrates our MaxFlow architecture. The MaxFlow system comprises four key component modules:

1) **Feature Extraction (FeExt):** In this initial stage, the network employs several residual stages to extract pertinent features from input image pairs.

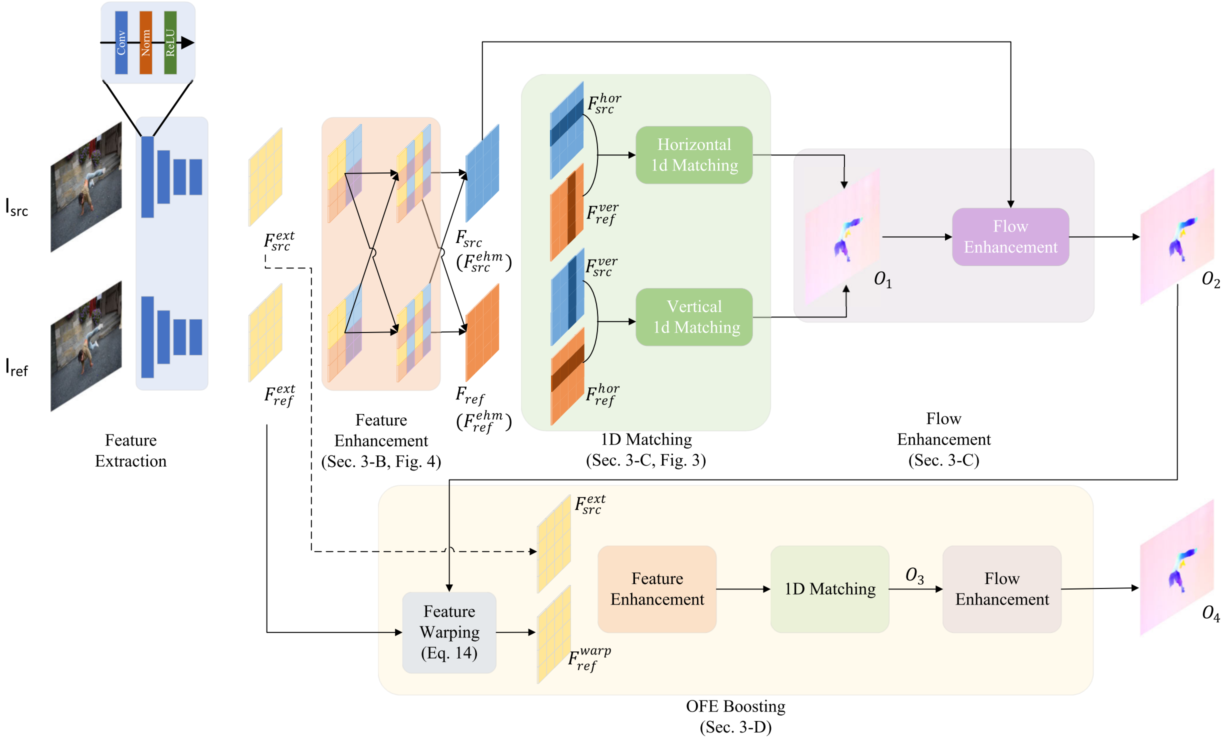

FIGURE 2. **Overall framework of our proposed network, MaxFlow. Our network consists of feature extraction, feature enhancement, 1D matching, flow enhancement, and optical flow estimation boosting.**

2) **Feature Enhancement (FeEhm):** Subsequently, the extracted features undergo an enhancement process to bolster their distinctiveness, thereby facilitating more effective OFE. Detailed elaboration on this process is presented in Section III-B.
3) **1D Matching (1DMat):** OFE is performed through a process, termed as 1D matching, a novel approach that addresses feature matching in 1-dimensional (horizontal and vertical) manners. This is very effective for low-complexity operations, which explicitly estimates the optical flows from 1D cost volumes at one single shot, rather than using iterative ConvGRU as in Flow1D [7].
4) **Flow Enhancement (FlEhm):** The final stage encompasses the application of flow propagation [6], [8] techniques, supplemented by a series of CNN layers, to refine the estimated optical flow.

The FeExt module consists of three cascaded residual blocks, which generates two 1/8-sized feature maps: one for the source image, F^{ext}_{src} and the other for the reference image, F^{ext}_{ref}. Further insights into the specific mechanisms and methodologies employed in the FeEhm, 1DMat, and FlEhm modules will be discussed in subsequent sections of this paper. Also, we will briefly explain our refinement stage for further performance improvement by repeating the FeEhm-1DMat-FlEhm operation, and the objective function used in our training.

A. 1D MATCHING

Before introducing our proposed 1D matching for correspondence finding with 1-D cost volumes, we briefly describe a global correspondence matching from which our 1D matching is motivated.

1) GLOBAL MATCHING

The principle behind *global matching*, proposed in GMFlow [6], involves comparing correlations between two input images, I_{src} and I_{ref}. The correlation, or cost volume, is then computed as follows:

$$C_{ij} = \frac{F_{src}(i,j)F^T_{ref}}{\sqrt{D}} \in \mathbb{R}^{H\times W}, \tag{1}$$

where F_{src} and F_{ref} represent the extracted features from I_{src} and I_{ref}, respectively, obtained via a convolutional neural network (CNN) encoder and additional feature enhancement layers. Also, (i,j) denotes the coordinates within F_{src}, D represents the channel size of F_{src} and F_{ref} which is used to normalize the cost volume, and H and W are the height and width of F_{src} and F_{ref}. C_{ij} symbolizes the cost volume corresponding to the (i,j) coordinate in F_{src}.

The cost volume C_{ij} in Eq 1 can take on negative and positive values. So, C_{ij} is probabilistically normalized as M_{ij}

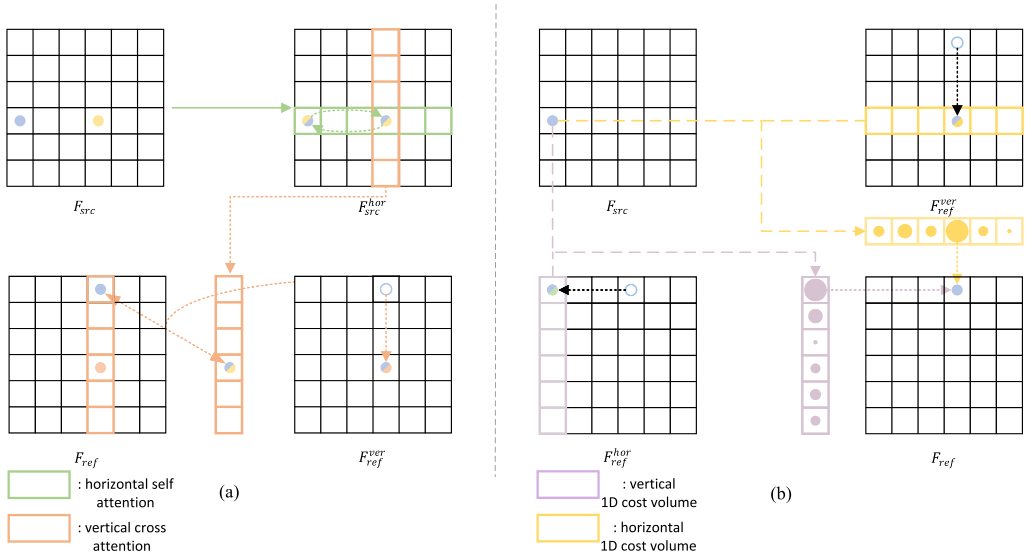
F_{src}
F_{src}^{hor}
F_{ref}
F_{ref}^{ver}
: horizontal self attention
: vertical cross attention
(a)
F_{src}
F_{ref}^{ver}
F_{ref}^{hor}
F_{ref}
: vertical 1D cost volume
: horizontal 1D cost volume
(b)

FIGURE 3. Our proposed 1D matching. (a)1D horizontal cost volume construction, following Flow1D [7]. Green line: horizontal self-attention, F_{src} horizontally exchange their features using horizontal self-attention. Orange line: vertical cross attention, F_{ref} and F_{src}^{hor} are vertically cross-attended to correctly exchange the vertical features in $F_r ef$ to highly correlated row. (b) Our proposed 1D matching strategy for correspondence matching. The horizontal and vertical cost volumes for a selected pixel in I_{src} are depicted at the bottom and left of I_{ref}, respectively. These cost volumes are derived from aggregated features, and the centroid of each cost volume is calculated to determine the corresponding x and y coordinates in I_{ref}.

between 0 and 1 as:

$$M_{ij} = Softmax(C_{ij}) \in \mathbb{R}^{H\times W}. \tag{2}$$

where $Softmax(\cdot)$ is the softmax operator. The correspondence for each feature in F_{src} is identified by applying the 2D grid to the normalized cost volume M_{ij} as:

$$c_{ij} = M_{ij}G \in \mathbb{R}^2, \tag{3}$$

where c_{ij} designates the corresponding pixel coordinates of I_{ref} at location (i, j) in I_{src}, and G is the matrix form of 2D coordinates. Optical flows are then determined by subtracting the coordinates of the original pixels in I_{src} from the corresponding coordinates in I_{ref}.

The formulation of OFE as a matching problem exhibits notable efficacy under the conditions where corresponding pixels are present in both the source and reference images. We introduce a novel formulation for OFE, termed as *1D Matching*, which is an essential low-complexity component of our proposed MaxFlow architecture. While global matching can predict optical flow without the need for GRU iterations, it is not feasible for lightweight optical flow estimation due to its high memory demand proportional to $(HW)^2$.

Consequently, we propose a 1D matching strategy that significantly reduces both memory and computation, scaling proportionally to $HW(H+W)$. For this purpose, we utilize the concept of 1D cost volume construction as introduced in [7]. The rest of this section provides a succinct overview of the 1D cost volume construction process, followed by a detailed explanation of our 1D matching strategy.

2) 1D COST VOLUME CONSTRUCTION [7]

Our proposed 1D matching, performed in feature domain, is based on the previous 1D cost volume construction. To construct the horizontal cost volume for our 1D matching approach, the objective is to ensure that the maximum value in the horizontal cost volume for each pixel in F_{src} corresponds to the x coordinate of the matched pixel in F_{ref}. This necessitates an aggregation of features along the same column in F_{ref}, so influencing the cost volume across different rows, even within a single horizontal cost volume construction. Fig. 3-(a) illustrates an example of a 1-D (horizontal) cost volume construction. We formulate this aggregation as a column-wise linear combination in F_{ref}:

$$F_{ref}^{ver}(h, w) = \sum_{i=0}^{H-1} f_{i,h,w}^{ver} F_{ref}(i, w), \tag{4}$$

where h and w range over [0, H-1] and [0, W-1] respectively, $F_{ref}^{ver}(h, w)$ represents the aggregated feature of F_{ref} along the column, and $f_{i,h,w}^{ver}$ are the weights of this linear combination.

The weight $f_{i,h,w}^{ver}$ plays a crucial role, indicating the significance of each pixel in I_{ref} for constructing the aggregated feature F_{ref}^{ver} at the coordinates (h, w). These weights are highly dependent on the features in I_{src} that share the same vertical coordinate h. To address this, they first aggregate F_{src} in a row-wise manner:

$$F^{hor}_{src}(h,w) = \sum_{i=0}^{W-1} Attn^{hor}(i,j)F_{ref}(h,i),$$
$$Attn^{hor}(i,h,w) = softmax\left(\frac{F_{src}(h,i)F_{src}(h,w)^T}{\sum_{j=0}^{W-1} F_{src}(h,j)F_{src}(h,w)^T}\right), \quad (5)$$

where $F^{hor}_{src}(h,w)$ denotes the aggregated F_{src} across the row. This step ensures that every feature in F_{src} contributes to the vertical cross-attention mechanism. The weight $f_{i,h,w}$ is then derived using the aggregated features F^{hor}_{src} and F_{ref} as:

$$f^{ver}_{i,h,w} = F_{ref}(i,w)F^{hor}_{src}(h,w)^T. \quad (6)$$

The horizontal correlation, or cost volume, is subsequently computed as follows:

$$C^{hor}(i,h,w) = \frac{1}{\sqrt{D}}F_{src}(h,w)\hat{F}_{ref}(h,i)^T, \quad (7)$$

where $C^{hor}(i,h,w)$ represents the horizontal cost volume corresponding to source features at (h,w) and reference features at the i-th column. Similarly, the vertical cost volume can be constructed using vertical 1D self-attention and horizontal 1D cross-attention.

3) 1D MATCHING

Fig. 3-(b) illustrates our 1D matching strategy for OFE. As shown in Fig. 3-(b), our 1D matching is performed on the constructed 1D cost volume in Fig. 3-(a), and is actually a dimensional modification of the global matching technique [6]. As observed, through the construction of a 1D horizontal cost volume, each pixel of the source image effectively exchanges information with its corresponding pixel in the reference image, localized within an appropriate row and column. This ensures that even if the corresponding pixel does not exist in the same row or column, its presence can still be considered within the 1D cost volume. Subsequently, the pixel-wise centers of gravity in the computed horizontal cost volume C^{hor} and vertical cost volume C^{ver} are assigned as the x and y coordinates of the corresponding pixels, respectively. Also, the cost volume C^{hor}_{ij} can take on negative and positive values. So, C^{hor}_{ij} is normalized as M^{hor}_{ij} using the *softmax* operator:

$$M^{hor}_{i,j} = softmax(C^{hor}(i,j,\cdot)) \in \mathbb{R}^W. \quad (8)$$

The horizontal correspondence for each feature in F_{src} is identified by applying the 1D horizontal grid to the normalized cost volume M^{hor}_{ij} as:

$$c^{hor}_{ij} = M^{hor}_{ij}G^{hor} \in \mathbb{R}, \quad (9)$$

where G^{hor} is an matrix form of 1-dimensional (horizontal) coordinates. The vertical correspondence pixel c^{ver}_{ij} can be found in a similar way. As a result, we can find 2D correspondence pixels c_{ij} as

$$c_{ij} = [c^{hor}_{ij}, c^{ver}_{ij}], \textit{and } o_{ij} = [c^{hor}_{ij} - i, c^{ver}_{ij} - j], \quad (10)$$

where o_{ij} indicates optical flow estimation at (i,j)-th feature location in the source image.

B. FEATURE ENHANCEMENT

While calculating the centroid of the cost volume is a viable approach for determining corresponding pixels, relying solely on features derived from a CNN network does not yield optimal results. This limitation arises due to the absence of discriminative differences in continuous patterns or homogeneous regions within the features. To overcome this, it is imperative to enhance the features using a vision transformer. In the work of Xu et al. [6], a Swin layer [13] was employed for feature enhancement, applying both self and cross attention mechanisms not only to enhance the features but also to facilitate inter-feature awareness.

However, large motion detection necessitates a corresponding increase in window size, leading to significantly higher computational and memory requirements, which scale with the square of the image resolution. To address these challenges, the MaxViT layer [14] is employed as an alternative to the Swin layer. As shown in Fig. 4, the MaxVit layer divides the image into both a local window and a global grid, thereby achieving a global receptive field while maintaining computational efficiency that scales linearly with image resolutions. This global receptive field is instrumental in detecting large motions between images and in enhancing the self-information of the features. We indicates enhanced features of source and reference image as F^{ehm}_{src} and F^{ehm}_{ref}, respectively. However, for simplicity, the superscript has been omitted in 1D matching section as F_{src} and F_{ref}.

Additionally, for the performance enhancement and model robustness to the changes in image resolutions, we adopted a method of applying continuous augmentation to positional encoding [22]. As the transformer structure is utilized in the feature enhancement module, we add positional encoding to the features obtained during the feature extraction phase. However, if training is only conducted at a fixed target resolution for each dataset and the actual inference is performed on images of different resolutions, the network will not have learned the positional encoding for those resolutions, resulting in lower performance. Following Cape [22], we apply global shift, local shift, and global scaling augmentations to the fixed grid used to create the sinusoidal positional encoding. The details are as follows.

Global shift hides the absolute positional information, but only considers relative relations. This transformation randomly shift grids horizontaly and verticaly with same magnitude $\Delta x, \Delta y \sim U(-\Delta_{max}, \Delta_{max})$

$$x'_i \leftarrow x_i + \Delta x, \quad y'_j \leftarrow y_j + \Delta y, \quad (11)$$

where $U(x,y)$ denotes uniform distribution between x and y. To prevent the network from memorizing the intervals of

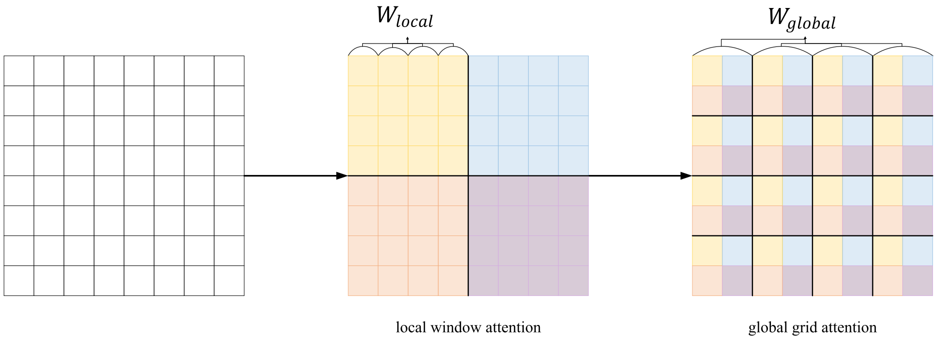

FIGURE 4. Feature enhancement layer. Our feature enhancement layer utilizes MaxViT layer, which consists of local window attention and global grid attention. Local window attention subdivide image with equal sized windows(window size is W_{local} in this figure) and apply self-attention to feautures in same image. In global grid attention, images are divided into fixed number of gird (W_{global} in the figure) and aggregate features in same position in each grid. Aggregated features are passed through self-attention layer to achieve large receptive field. The features that are included in the same self-attention are represented by the same color in the figure.

the positional encoding, we further apply local shifts and global scaling. By shifting the positional encoding through a global shift, the network is trained on a variety of positional encodings, yet the interval between each encoding remains consistent. This consistency means that when the inference resolution increases or decreases, the rate of change within the positional encoding window will differ from that during training, which can also reduce inference performance. To resolve this, we apply a local shift that shifts each grid differently, and a global scaling that further augments the intervals between grids. The local shifts in horizontal and vertical directions at location (i, j) are obtained using sampled noise $\varepsilon_{x,i} \sim U(-\varepsilon_{x,max}, \varepsilon_{x,max})$ and $\varepsilon_{y,j} \sim U(-\varepsilon_{y,max}, \varepsilon_{y,max})$ as:

$$x'_i \leftarrow x_i + \varepsilon_{x,i}, \quad y'_j \leftarrow y_j + \varepsilon_{y,j} \tag{12}$$

To prevent the memorization of distances between positional embedding grids, we also introduce a random global scale λ from $\log \lambda \sim U(-\log \lambda_{max}, \log \lambda_{max})$ as:

$$x'_i \leftarrow \lambda x_i, \quad y'_j \leftarrow \lambda y_j \tag{13}$$

In our model, Δ_{max}, $\varepsilon_{x,max}$, $\varepsilon_{y,max}$, and λ_{max} are set to 0.5, $1/W$, $1/H$, and 1.4, respectively.

C. FLOW ENHANCEMENT

The flow enhancement module in our approach is divided into two key components: flow propagation and flow head.

1) FLOW PROPAGATION

The accuracy performance of the 1D matching module is contingent on the existence of the corresponding pixels within the reference image for each pixel in the source image. In real-world scenarios, occlusions and out-of-boundary movements often disrupt the existence of correspondences. To ensure accurate flow estimation for occluded pixels, our MaxFlow adopts a flow propagation layer from the previous works [6], [8]. This layer extends the optical flow to the areas lacking pixel correspondences, aiming to replicate the actual flow obscured by occlusions or boundary constraints. The flow propagation layer measures the self-similarity of F_{src}, and propagates the highly accurate flows for which the correspondences exist, to a region without correspondences. The self-similarity-based flow propagation is performed under the assumption that two highly similar regions are in the same context or object, and their flows would be similar. Our MaxFlow, differentiated from that in [6], constrains the flow propagation within a local window, significantly reducing computational load while maintaining the prediction accuracy.

2) FLOW HEAD

The flow head component serves to refine the optical flows derived from the flow propagation. It consists of several CNN layers that further enhance the flows. Unlike the 1D matching and flow propagation that operate in the coordinate domain, the flow head addresses the continuity of optical flows within a same object. It processes the optical flows from the flow propagation stage in conjunction with enhanced source image features, culminating in an optical flow representation that more accurately reflects its inherent characteristics.

D. OFE-BOOSTING

The MaxFlow incorporates an OFE-boosting stage to enhance the prediction accuracy of OFE by repeating the feature enhancement, 1D matching and flow enhancement steps one more time, similar to the work [6]. This OFE-boosting first involves the warping of the reference image features, F^{ext}_{ref} onto the source image features by the preliminary optical flow (O_2 in Fig. 2) by

$$F^{warp}_{ref} = OFW(F^{ext}_{ref}, O_2), \tag{14}$$

where $OFW(\cdot)$ indicates a backward warping operation [23] that brings the corresponding reference feature elements into closer proximity of the source feature elements.

Such proximity facilitates more effective information transfer between these elements, thereby contributing to the prediction of more refined optical flows. These warped features then undergo the feature enhancement, 1D matching, and flow enhancement in order.

E. LOSS

The MaxFlow framework initially derives optical flows through 1D matching in its primary phase. These flows are subsequently refined through the 1D matching and the flow enhancement two times (See Fig. 2) and another two times for the OFE-boosting stage. While it is feasible to supervise solely the final output (optical flow estimates) for simplicity, supervising each stage of optical flow generation ensures alignment with our network's objectives and promotes stable training. The supervision is implemented as follows: the optical flows that are the outputs of the 1D matching and the flow enhancement in both initial and OFE-boosting stages are compared against the ground truth optical flows to calculate the loss for training. This is quantified using the total loss function $\mathcal{L}$:

$$\mathcal{L} = \sum_{i=1}^{4} \gamma^{4-i} \|\mathbf{O}_{gt} - \mathbf{O}_i\|_1 \tag{15}$$

where γ denotes a scaling factor applicable to each stage of OFE, $\mathbf{O}_{gt}$ is the ground truth optical flow, and $\mathbf{O}_i$ indicates the optical flow map obtained at each respective stage such as the outputs of the first 1-D matching, first flow enhancement and second 1-D matching and second flow enhancement modules for $i = 1, 2, 3$ and 4, respectively, as shown in Fig. 2.

IV. EXPERIMENTS

A. DATASETS

In alignment with established protocols in [5] and [6], our approach involves training the proposed MaxFlow model using the FlyingChairs [2], FlyingThings3D [24], and HD1K [25] datasets. Evaluation is conducted on the Sintel [26] and KITTI [27] datasets. The FlyingChairs dataset, a synthetic collection, is composed of 3D-rendered images of flying chairs, supplemented by the images from various open-source databases. These chairs undergo random affine transformations to simulate real-world motion. In contrast, FlyingThings3D, another synthetic dataset, includes 3D objects following 3D trajectories, unlike the 2D motion in FlyingChairs. Sintel, which is also synthetic, features more lifelike motion derived from an actual animated film. HD1K and KITTI, sourced from real-world environments, comprise images captured from vehicles, and feature the sparse optical flows calculated using external sensors. Given the limited number of training images in the test datasets, we initially train our model on the FlyingChairs and FlyingThings3D datasets, followed by fine-tuning it on Sintel, HD1K, and KITTI datasets. The other models under comparison are also trained and tested in the same way as our MaxFlow model.

B. IMPLEMENTATION DETAILS

The implementation of our MaxFlow model is based on the foundational code of [6], utilizing Pytorch, with CUDA for efficient memory handling during flow propagation. Training commences on the FlyingChairs dataset for 100k iterations, followed by 800k iterations on the FlyingThings3D dataset. Subsequent fine-tuning involves a mix of data from Sintel, FlyingThings, KITTI, and HD1K datasets over 200k iterations, preparing the model for evaluation on Sintel. For KITTI-based testing, the Sintel-optimized model undergoes an additional 100k iterations of training on the KITTI dataset. The training learning rates are initially set to 4×10^{-4}, 2×10^{-4}, 2×10^{-4} and 1×10^{-4} for FlyingChairs, FlyingThings3D, Sintel and KITTI datasets, respectively, with a decay implemented via a CosineAnnealing scheduler. The AdamW optimizer is employed, with parameters (β_1, β_2) set to (0.9, 0.9) and a weight decay of 1×10^{-8}. To ensure stable learning, gradient clipping is applied with a clipping parameter of 1. Data augmentation techniques include random resizing and cropping to dimensions (384, 512), (384, 768), (320, 896), and (320, 1152), as well as random horizontal flips, color jitters, and grayscale conversions to diversify the training dataset. Full training is achieved in two and a half days using two NVIDIA RTX 3090 GPUs. The pretrained model based on FlyingChairs and FlyingThings3D datasets is referred to as the *things* model, whereas the fine-tuned models for the Sintel and KITTI datasets are denoted as the *sintel* and *kitti* models, respectively.

C. EXPERIMENT RESULTS

1) EVALUATION METRICS AND SCENARIOS

To objectively assess the prediction accuracy performance, we use (i) end-point error (EPE) between estimated flow and ground truth flow, (ii) F1-all, which indicates f1 score where a pixel being deemed accurately estimated if the error in flow is less than 3 pixels or less than 5% of its ground truth value. The prediction accuracy performance is evaluated in several scenarios in terms of motion amounts: 's0-10', 's10-40' and 's40+' indicates the evaluations of the trained methods for the separated datasets of average per-pixel motion magnitudes in the range between 0 and 10, the range between 10 and 40, and the range of 40 and above. 'all' implies the total range of the three.

2) QUALITATIVE RESULTS

Figure 5 shows subjective OFE comparison of our MaxFlow and other SOTA methods. As shown, our MaxFlow alongside the matching-based method GMFlow [6] demonstrates precise OFE in ambiguous regions (the bott om left part of the first-column OFE maps in Figure 5), and in the fine-detail regions (the fingers and box objects of the second-column OFE maps in Figure 5). These results demonstrate the superior accuracy of the matching-based methods compared to the regression methods such as RAFT [5] and Flow1D [7].

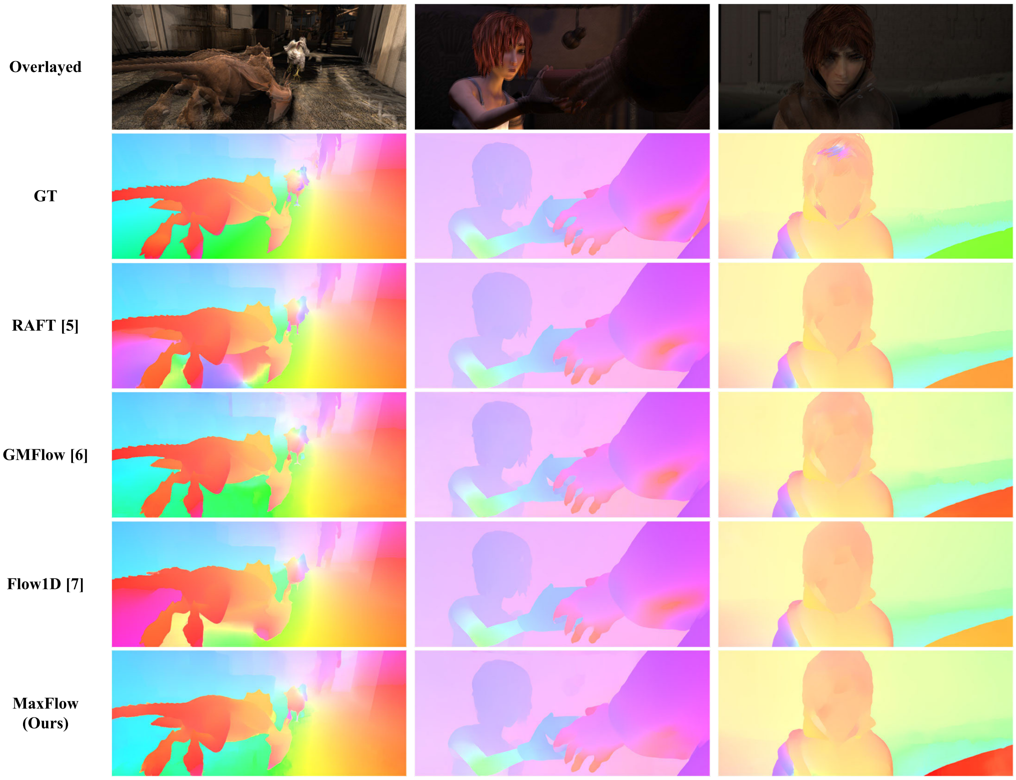

FIGURE 5. Qualitative comparison on Sintel test datasets.

TABLE 1. Performance comparison of *things* methods in terms of the prediction accuracy (F1-epe and F1-all), the number of parameters and the multiplication and accumulation calculations (MACs). Note that *things* methods indicates that all the models under comparison are trained with FlyingChairs and FlyingThings3D datasets, and then tested with Sintel train and KITTI train datasets. Bold: best, underline: second-best.

		Sintel (train)		KITTI-15 (train)			Complexity			
							448×1,024		1,080×1,920	
Strategies	Methods	*Clean* (EPE)	*Final* (EPE)	F1-EPE	F1-all	#'s of parms. (M)	iters 12 or w/o boost (GMACs)	iters 31 or w/ boost (GMACs)	iters 12 or w/o boost (GMACs)	iters 31 or w/ boost (GMACs)
High Performance	RAFT [5]	1.43	2.71	5.04	17.40	5.26	376.2	801.5	1,927.0	3,864.0
	FlowFormer [17]	**1.01**	**2.40**	**4.09**	**14.72**	16.17	662.6	1,155.0	OOM	OOM
	GMFlow [6]	1.08	2.48	7.77	23.40	4.83	203.6	499.8	2,421.0	5,047.0
Low Complexity	SCV [9]	1.29	2.95	6.80	19.30	5.24	1,181.0	2,834.0	OOM	OOM
	Flow1D [7]	1.98	3.27	6.69	22.95	5.73	364.2	782.7	1,666.0	3,572.0
	MaxFlow (ours)	1.51	3.28	9.98	32.70	**4.20**	**106.1**	**145.0**	**486.9**	**668.3**

3) QUANTITATIVE RESULTS

Table 1 compares the the results of the *things* methods in terms of the prediction accuracy, the number of parameters and the multiplication and accumulation calsulations (MACs). Note that the *things* methods were trained on the FlyingChairs and FlyingThings3D datasets, and then tested on the Sintel train and the KITTI-15 train datasets. As shown in Table 1, FlowFormer [17], which is targeted for high performance, achieves the highest performance in terms of prediction accuracy. However, it consumes the second-highest MACs at the 448 × 1.024 resolution (662.6 GMACs for 12 iterations and 1,155 GMACs for 31 iterations). On the other hand, our MaxFlow consumes the lowest MACs, showing the accuracy performance superior to Flow 1D [7] but inferior to SCV [9] that is even designed for low complexity but consumes about 10× to 20× MACs. It is noted in Table 1 that the increase in computation due to the refinement (boosting) by our MAXFlow was very small compared to the GRU-based iterative methods such as RAFT [5], FlowFormer [17] and Flow1D [7]. Especially, our FlowMax with refinement only takes a 40% computational complexity of Flow1D [7] at both 448 × 1.024 and 1,080 × 1.920 resolutions after 12 iterations.

As extensive experiments, the pretrained *things* methods were further fine-tuned on Sintel dataset, which are referred to as *sintel* methods, and on KITTI train dataset, which are

TABLE 2. Performance comparison of *sintel* and *kitti* methods in terms of the prediction accuracy. Note that *sintel* and *kitti* methods indicates that all the models under comparison are finetuned on Sintel and KITTI datasets from the *things* methods, respectively, and then they are all tested with Sintel test and KITTI test datasets. Bold: best, underline: second-best. Note that SCV outperforms our MaxFlow in EPE but it is 10 to 20 times complex than MaxFlow in MACs.

Strategies	Methods	Sintel (test)								KITTI (test)
		Clean (EPE)				*Final* (EPE)				F1-all
		all	s_{0-10}	s_{10-40}	s_{40+}	*all*	s_{0-10}	s_{10-40}	s_{40+}	
High Performance	GMFlow [6]	1.736	0.499	$\underline{0.971}$	9.724	2.902	0.714	$\underline{1.659}$	16.751	9.32
	RAFT(2-view) [5]	1.940	-	-	-	3.18	-	-	-	$\underline{5.10}$
	RAFT(warm-start) [5]	$\underline{1.609}$	$\underline{0.341}$	1.036	$\underline{9.288}$	$\underline{2.855}$	$\underline{0.634}$	1.823	$\underline{16.371}$	-
	FlowFormer [17]	**1.159**	**0.259**	**0.822**	**6.435**	**2.088**	**0.459**	**1.472**	**11.656**	**4.68**
Low Complexity	SCV [9]	**1.720**	**0.29**	**0.911**	$\underline{10.782}$	$\underline{3.603}$	0.787	$\underline{2.11}$	$\underline{21.197}$	**6.17**
	Flow1D [7]	2.238	0.509	1.372	12.889	3.806	$\underline{0.738}$	2.479	22.221	$\underline{6.27}$
	MaxFlow (ours)	$\underline{1.868}$	$\underline{0.45}$	$\underline{1.176}$	**10.566**	**3.295**	**0.663**	**1.898**	**19.729**	10.74

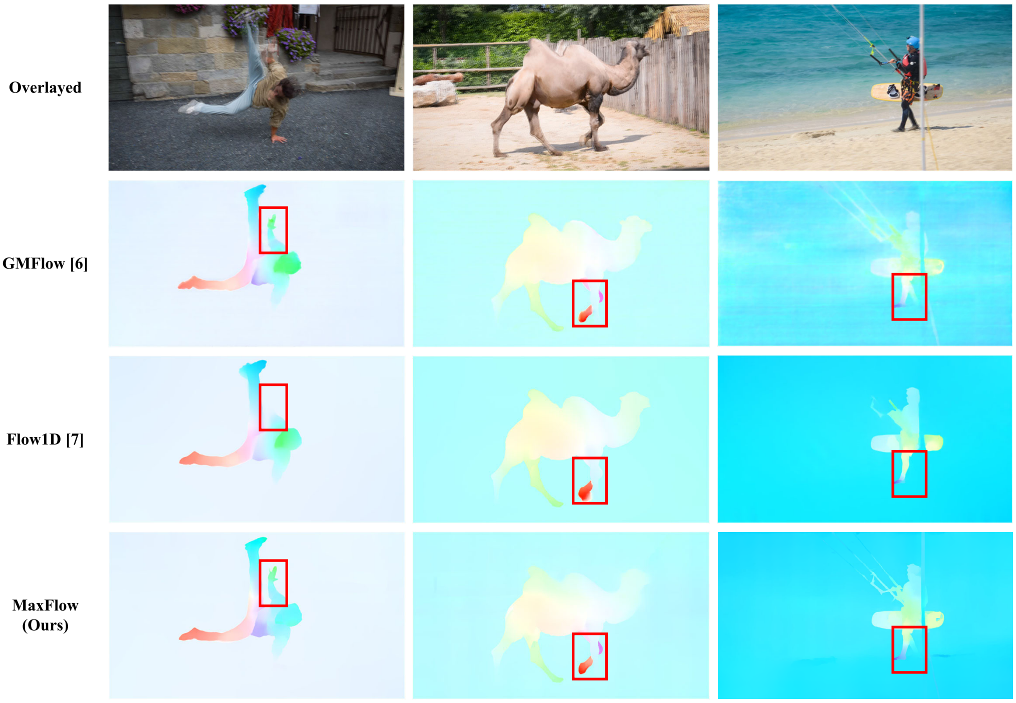

FIGURE 6. Subjective OFE Comparison on 2K images in Davis dataset.

denoted as *kitti* methods. *sintel* and *kitti* methods are tested on Sintel and KITTI test datasets, respectively. Table 2 shows the performance comparisons for *sintel* and *kitti* methods. As shown in Table 2, our MaxFlow exhibited a comparable accuracy performance against SCV [9], yielding lower and higher metrics for *Clean*, and *Final* Sintel test datasets, respectively. On the KITTI dataset, it should be noted in Table 2 that our MaxFlow shows lower performance, especially compared to the regression-based methods such as RAFT [5], Flow1D [7], SCV [9], and FlowFormer [17]. The matching-based methods such as our MaxFlow and GMFlow [6] are relatively weak in handling occlusions that extend outside the images, which is a common occurrence in KITTI dataset. It is noted that GMFlow [6] has recently extended to overcome this issue by applying a computationally expensive iterative GRU-based regression refinement module after global matching in Unimatch [10], thus *dramatically* increases the computation complexity.

Next, we evaluate our MaxFlow and the SOTA methods on high-resolution images of 1080p to 2K (1,280 × 720 to 1,920 × 1.080) size. The *sintel* methods were further fine-tuned with an additional 100k iterations to enhance performance using the Spring dataset [28], which is composed of 2K images tailored for high-resolution OFE. For validation, we employ the DAVIS dataset [29], a benchmark for video segmentation that includes video sequences of 2K resolutions. Figure 6 depicts the subjective OFE results for our MaxFlow and the SOTA methods. Note in Figure 6 that the two input frames in the first row are overlaid to see the motion amounts between them. As clearly shown in the red boxes of Figure 6 that our MaxFlow accurately captures large motion (such as the arm of the B-boy in the left example)

TABLE 3. Ablation study on the impact of individual components (our design choices) within the MaxFlow architecture. Bold: best, underline: second-best.

	Things	Sintel (train)							
		Clean (EPE)				*Final* (EPE)			
	EPE	all	s_{0-10}	s_{10-40}	s_{40+}	all	s_{0-10}	s_{10-40}	s_{40+}
Ours w/o maxvit	4.701	1.789	0.417	1.980	11.1	3.342	0.689	3.962	20.8
Ours w/o 1dm	4.484	**1.661**	**0.392**	**1.811**	**10.34**	3.268	**0.644**	**3.612**	21.13
Ours w/o cape	**4.435**	1.8	0.46	1.935	11.02	3.266	0.732	3.943	19.76
Ours	4.546	1.769	0.425	1.984	10.83	**3.075**	0.657	3.875	**18.48**

and detailed optical flow structures (the camel's leg in the middle example). Furthermore, when GMFlow [6] is applied to 1600 × 900 resolution image (right column of Fig. 6), it produces artifacts due to its poor robustness in inference resolution change. On the other hand, our MaxFlow generates optical flow estimation without artifact and more accurate details (right leg of person highlighted in red box) than Flow1D.

4) ABLATION STUDY

We perform ablation study on the impact of individual components (our design choices) within the MaxFlow architecture. For this, we denote the variants of our MaxFlow as following: (i) 'Ours w/o maxvit' as the scenario where the MaxViT [14] layer, used in the feature enhancement layer, is replaced by a Swin [13] block; (ii) 'Ours w/o 1dm' that signifies the use of a global matching strategy in place of 1D matching; (iii) 'Ours w/o cape' that denotes the condition where the continuously augmented positional encoding (Cape) [22] is omitted from the MaxFlow. The three variants are trained using the FlyingChairs dataset for 100k iterations and the FlyingThings3D dataset for 200k iterations. Table 3 shows the accuracy performance comparison of OFE for the three variants of our MaxFlow. As shown in Table 3, 'Ours w/o maxvit' leads to a performance decline of approximately 0.27 for the Sintel train *Final* datasets. This effect is more prominent in the s_{40+} scenario, where we observe a decrease of about 2.32 in terms of EPE. It can be noted for 'Ours w/o maxvit' in Table 3 that the restricted receptive field of the Swin layer is a critical factor, particularly affecting the EPE performance for the long-range motion scenario (s_{40+}). Moreover, 'Ours w/o 1dm' results in a slight enhancement in performance, with an increase of 0.06 and 0.1 in the FlyingThings validation and sintel (train) *Final* datasets, respectively. However, this benefit comes at a significant cost increasement in terms of memory consumption from 4,013MB to 11,631MB, which escalates by approximately 289% for input resolution of 1,920 × 1.080. This finding underscores the practical balance between performance and efficiency offered by 1D matching. Additionally, 'Ours w/o cape' demonstrates high performance on the FlyingThings3D dataset, which was used during training. However, on the Sintel dataset that features higher resolutions not utilized during training, 'Ours w/o cape' exhibits inferior performance. This outcome aligns with our expectation on Cape [22] that helps enhancing the model's generalization capabilities.

V. CONCLUSION

In this paper, we propose a novel lightweight optical flow estimation network, MaxFlow, which utilizes MaxViT and 1D matching. MaxViT enables our network to have both a large receptive field at an efficient computational cost. We also propose a novel 1D matching strategy which explicitly inferences the correspondence between two input images with their 1D cost volume. Through these contributions, our MaxFlow can achieve superior optical flow estimation accuracy than a SOTA lightweight optical flow estimation network, Flow1D in various test dataset, such as Sintel, only with 40% of computational cost compared to Flow1D, and show comparable prediction performance at a significantly lower computation complexity (about 10× to 20 × lower) than another SOTA light weight method such as SCV.

REFERENCES

[1] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[2] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1647–1655.

[3] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.

[4] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 548–558.

[5] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. 16th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Aug. 2020, pp. 402–419.

[6] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao, "GMFlow: Learning optical flow via global matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8111–8120.

[7] H. Xu, J. Yang, J. Cai, J. Zhang, and X. Tong, "High-resolution optical flow from 1D attention and correlation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10478–10487.

[8] S. Jiang, D. Campbell, Y. Lu, H. Li, and R. Hartley, "Learning to estimate hidden motions with global motion aggregation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9752–9761.

[9] S. Jiang, Y. Lu, H. Li, and R. Hartley, "Learning optical flow from a few matches," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16587–16595.

[10] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, F. Yu, D. Tao, and A. Geiger, "Unifying flow, stereo and depth estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13941–13958, Nov. 2023.

[11] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[13] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[14] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, "MaxViT: Multi-axis vision transformer," in *Proc. 17th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 459–479.

[15] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, nos. 1–3, pp. 185–203, Aug. 1981.

[16] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2720–2729.

[17] Z. Huang, X. Shi, C. Zhang, Q. Wang, K. C. Cheung, H. Qin, J. Dai, and H. Li, "FlowFormer: A transformer architecture for optical flow," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 13677. Cham, Switzerland: Springer, 2022, pp. 668–685.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[20] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[21] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 22–31.

[22] T. Likhomanenko, Q. Xu, G. Synnaeve, R. Collobert, and A. Rogozhnikov, "CAPE: Encoding relative positions with continuous augmented positional embeddings," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 16079–16092.

[23] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[24] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4040–4048.

[25] D. Kondermann, R. Nair, K. Honauer, K. Krispin, J. Andrulis, A. Brock, B. Güssefeld, M. Rahimimoghaddam, S. Hofmann, C. Brenner, and B. Jähne, "The HCI benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 19–28.

[26] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2012, pp. 611–625.

[27] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3061–3070.

[28] L. Mehl, J. Schmalfuss, A. Jahedi, Y. Nalivayko, and A. Bruhn, "Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 4981–4991.

[29] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 724–732.

WONYONG SEO received the B.S. and M.S. degrees from the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2021 and 2023, respectively, where he is currently pursuing the Ph.D. degree. His research interests include low-level computer vision, especially in video, such as optical flow estimation, video frame interpolation, video restoration, generative models on the video domain, and deep learning.

WOONSUNG PARK received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2015, 2017, and 2021, respectively. He is currently a Staff Engineer with the Visual Display Business, Samsung Electronics, Suwon-si, South Korea. His current research interests include deep learning-based optical flow estimation, video frame interpolation, video super-resolution, and low-complexity deep learning architectures for on-device AI chips.

MUNCHURL KIM (Senior Member, IEEE) received the B.E. degree in electronics from Kyungpook National University, Daegu, South Korea, in 1989, and the M.E. and Ph.D. degrees in electrical and computer engineering from the University of Florida, Gainesville, in 1992 and 1996, respectively. He joined the Electronics and Telecommunications Research Institute, Daejeon, South Korea, as a Senior Research Staff Member, where he led the Realistic Broadcasting Media Research Team. In 2001, he joined the School of Engineering, Information and Communications University (ICU), Daejeon, as an Assistant Professor. Since 2009, he has been with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, where he is currently a Full Professor. He has published 180 (165) international (domestic) journal and conference papers. He holds several essential HEVC, VP/AV1, and VVC patents, and about 220 registered domestic and international patents in the areas of image restoration and video coding. His research interests include image restoration with deep learning, image-based 3D scene reconstruction, video coding, image analysis/understanding for natural images and video, and aerial and satellite images. He received a commendation from Korean President on the 54th National Innovation Day in 2019 and was awarded the Grand Prize for Research Excellence in Commemoration of the 50th Anniversary of the Founding of KAIST. He was invited to give a keynote speech on the evolution of conventional and deep video compression technologies at the 2020 Multimedia Modeling Conference. His team was awarded the Runner-Up in Challenge on the Video Temporal Super-Resolution Track in Advances in Image Manipulation (AIM) Workshop and challenges on image and video manipulation in ECCV 2020 and received the Winner Award on "PIRM Challenge on Perceptual Image Enhancement-Track A: Image Super-Resolution" in ECCV 2018. He has served as the Technical Program Chair for the Visual Communications and Image Processing (VCIP) 2023 and the ACM Multimedia Asia 2023.