2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

DenVisCoM: Dense Vision Correspondence Mamba for Efficient and Real-time Optical Flow and Stereo Estimation

Tushar Anand*, Maheswar Bora*, Antitza Dantcheva†, Abhijit Das*

***Abstract*— In this work, we propose a novel Mamba block DenVisCoM, as well as a novel hybrid architecture specifically tailored for accurate and real-time estimation of optical flow and disparity estimation. Given that such multi-view geometry and motion tasks are fundamentally related, we propose a unified architecture to tackle them jointly. Specifically, the proposed hybrid architecture is based on DenVisCoM and a Transformer-based attention block that efficiently addresses real-time inference, memory footprint, and accuracy for at the same time for joint estimation of motion and 3D dense perception tasks. We extensively analyze the benchmark trade-off of accuracy and real-time processing on a large number of datasets. Our experimental results and related analysis suggest that our proposed model can accurately estimate optical flow and disparity estimation in real time. All models and associated code are available at https://github.com/vimstereo/DenVisCoM.**

I. Introduction

Estimating optical flow and stereo disparity are fundamental challenges in computer vision, crucial for applications ranging from robotics to autonomous driving [1]. Early approaches, often relying on patch matching and assuming perfect lens rectification [2], have been superseded by deep learning methods. While CNNs [3], [4] and transformers [5] offer improved accuracy, they face a significant trade-off between computational efficiency and performance, hindering real-time deployment, particularly on resource-constrained platforms. Transformers, known for their ability to capture long-range dependencies, suffer from quadratic computational complexity, making them expensive to train and deploy.

State Space Models (SSMs), particularly Mamba [6] and its successors [7], present a promising alternative. SSMs achieve linear complexity and efficient parallel training, potentially resolving the accuracy-efficiency dilemma. However, directly applying Mamba, originally designed for sequential data, to vision tasks is non-trivial. Adaptations like Vision Mamba (ViM) [8], MambaVision [9], VMamba [10], and NC-SSD [7] address this by incorporating positional awareness and modified scanning strategies. These primarily focus on single-image tasks. Emerging multi-modal Mamba models [11], [12], [13] exist, but they do not directly address the core requirement of visual dense correspondence matching between image pairs, crucial for optical flow and stereo disparity. In this direction, very recently, ViM-disparity was proposed by [14], employing ViM for accurate and real-time disparity map estimation. However, dense correspondence tasks demand an explicit comparison of corresponding features, necessitating feature representations that encode both global position and local relationships. Existing approaches lack this nuanced correspondence handling.

Hence, in this paper, we introduce a novel hybrid architecture, optimized for optical flow and stereo disparity, that fundamentally rethinks the Mamba block. Our central innovation is a visual correspondent mechanism that fuses image pair features patch-wise within the Mamba sequence transformation. To enable robust fusion and capture essential relationships, we integrate self- and cross-attention, addressing Mamba's inherent lack of cross-correspondence. This builds upon the demonstrated success of hybrid SSM-transformer architectures in vision.

Our key contributions are:

- A novel hybrid SSM architecture combining Mamba with self- and cross-attention, specifically tailored for accurate and efficient optical flow and stereo disparity estimation.
- A redesigned Mamba block that facilitates joint learning of image pair features via a visual correspondence mechanism within the sequence transformation.

II. Related Work

A. Mamba Architectures

Mamba [6], [15] emerged as a state-space model (SSM) offering a computationally efficient alternative to transformers, reducing complexity from quadratic to linear while maintaining competitive performance in language modeling tasks. This success prompted efforts to adapt Mamba to vision. Key challenges included Mamba's unidirectional nature and lack of inherent positional awareness.

Vision Mamba (Vim) [8] addressed these by incorporating bidirectional SSMs and position embeddings. However, bidirectionality can introduce latency. MambaVision [9] adopted a hybrid approach, combining modified Mamba blocks with transformer blocks. VMamba [10] introduced a cross-scanning mechanism to improve information flow. Mamba-2 [7], based on the state space duality (SSD) framework, refined the selective SSM. VSSD [16] further enhanced this with a non-causal block (NC-SSD), maintaining global receptive fields and linear complexity while improving training and inference.

Beyond single-image tasks, Mamba has been explored for multi-modal applications, including medical image fusion (MambaDFuse) [11], multi-instance learning [12], object

*Machine Intelligence Group, Birla Institute of Technology and Science, Pilani, Hyderabad Campus, India.
†Université Côte d'Azur, Inria, France.
Email: abhijit.ads@hyderabad.bits-pilani.ac.in

detection [13], and the fusion of multispectral and RGB images [17]. However, these existing multi-modal Mamba models do not directly address the specific requirements of optical flow and stereo disparity estimation, particularly the need for a robust mechanism to establish and leverage visual dense correspondence between image pairs. Our work fills this gap by introducing a novel Mamba block architecture specifically designed for this purpose.

III. Proposed Methodology

A. Preliminaries

Mamba is based on SSMs that map a 1-D function $x(t) \in \mathbb{R} \rightarrow y(t) \in \mathbb{R}$ through a hidden state $h(t) \in \mathbb{R}^N$. It formulates $A \in \mathbb{R}^{N\times N}$ as the evolution parameter, and $B \in \mathbb{R}^{N\times 1}$ and $C \in \mathbb{R}^{1\times N}$ as projection parameters:

$$h(t) = Ah(t-1) + Bx(t), \quad y(t) = Ch(t) \tag{1}$$

Mamba are the discrete versions of the continuous system or SSMs, that include a timescale parameter Δ to transform the continuous parameters A and B to discrete parameters $\overline{A}$ and $\overline{B}$. The commonly used technique for this transformation is zero-order hold (ZOH). After the discretization of $\overline{A}$ and $\overline{B}$, the discretized version of the above equation using a step size of Δ is:

$$h_t = \overline{A}h_{t-1} + \overline{B}x_t, \quad y_t = Ch_t. \tag{2}$$

At last, the models compute output through a global convolution

$$\overline{K} = (C\overline{B}, C\overline{AB}, \ldots, C\overline{A}^{M-1}\overline{B}), \quad y = x * \overline{K}, \tag{3}$$

where M is the length of the input sequence x, and $\overline{K} \in \mathbb{R}^M$ denotes a structured convolutional kernel. To extend Mamba for the vision task, MambaVision modified the Mamba block. Assuming an input $X(t) \in \mathbb{R}^{T\times C}$ with sequence length T with embedding dimension C, the following equations describe the interaction between the Mixer and MLP blocks in MambaVision, where feature normalisation is followed by a Mixer and MLP block to process and combine the input representations:

$$\hat{X}^n = \text{Mixer}(\text{Norm}(X^{n-1})) + X^{n-1} \tag{4}$$

$$X^n = \text{MLP}(\text{Norm}(\hat{X}^n)) + \hat{X}^n. \tag{5}$$

The MambaVision Mixer operation combines sequential and spatial information using two branches, Scan *i.e.,* denoted as SSM and Convolution, and concatenates the results X_{out}.

$$X_1 = \text{SSM}(\sigma(\text{Conv}(\text{Linear}(C, C/2)(X_{\text{in}})))), \tag{6}$$

$$X_2 = \sigma(\text{Conv}(\text{Linear}(C, C/2)(X_{\text{in}}))), \tag{7}$$

$$X_{\text{out}} = \text{Linear}(C/2, C)(\text{Concat}(X_1, X_2)). \tag{8}$$

Each output is projected to be half the size of the original embedding dimension to maintain a similar number of parameters to the original block.

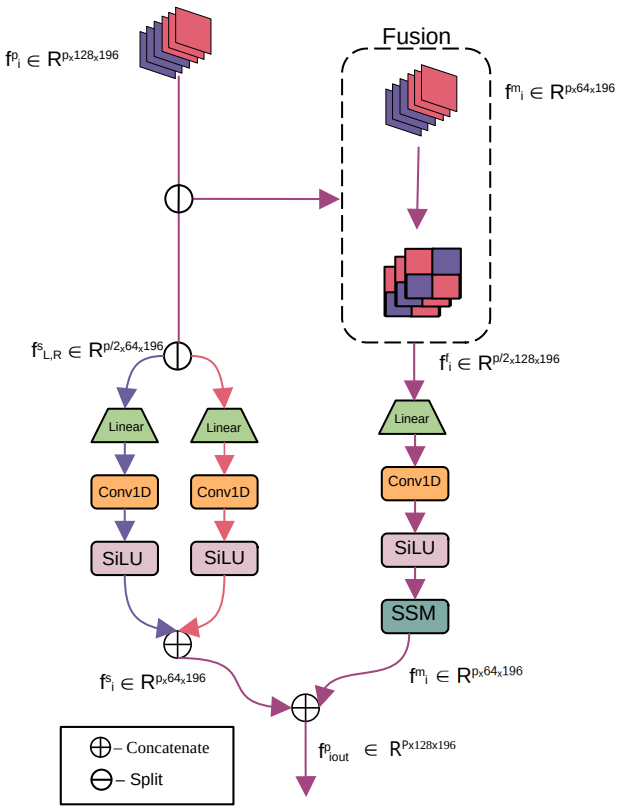

Fig. 1. The proposed modified DenVisCoM block for sequential joint modeling in the sequence transformation block of Mamba.

B. Proposed DenVisCoM Block

The proposed Mamba block, DenVisCoM, aims to model the dense vision correspondence between a pair of features by jointly learning the pair of input images in the sequence transformation pipeline (See Figure 1). To achieve this, we introduce two symmetrical convolution branches for each input feature and one joint Scan branch, which takes block-wise fused features (as in the fusion block of Figure 1). Each convolution branch consists of a linear projection layer, a Conv1D layer, and a SiLU activation function. The purpose of these symmetric paths is to independently compensate for any information loss in both the left and right images caused by the SSM branch due to its sequential nature.

$$X_L = \sigma(\text{Conv1D}(\text{Linear}(f^s_{iL}))) \tag{9}$$

$$X_R = \sigma(\text{Conv1D}(\text{Linear}(f^s_{iR}))) \tag{10}$$

The Scan branch, *i.e.,* the SSM branch, consists of a linear projection layer, a Conv1D layer, a SiLU activation function, and an SSM for handling long-range dependencies jointly for both inputs while preserving locality with convolutional layers.

$$X_{\text{SSM}} = \text{SSM}(\sigma(\text{Conv1D}(\text{Linear}(\text{Fusion}(f^m_i))))) \tag{11}$$

Further, the output of each convolution branch is concatenated, followed by further concatenation with the Scan

branch to obtain the output:

$$X_{\text{out}} = \text{Concat}(X_{\text{SSM}}, \text{Concat}(X_L, X_R)). \quad (12)$$

C. Proposed Architecture

To adapt and efficiently extract features via Mamba, the proposed model passes the input images via CNN and then to the Mamba block in the form of patches (See Figure 2). Specifically, the proposed model is a hybrid SSM-based approach, which consists of SSM and Transformer-based attention blocks, as in MambaVision. Precisely, in our proposed architecture, feature normalization is followed by SSM and attention blocks in order to process and combine the dense correspondence of input representations of a pair of input images. We proceed to explain the architecture in detail.

The two left and right (for stereo disparity estimation task) or consecutive frames (for flow) as images ($f_{INPL}, f_{INPR} \in \mathbb{R}^{3\times m\times n}$) are passed through two separate ResNet18 [18]-based CNN encoders to extract their corresponding $8\times$ downsampled feature representations ($f^L, f^R \in \mathbb{R}^{128\times \frac{m}{8}\times \frac{n}{8}}$), where m and n represent the spatial dimensions.

$$f^L = \text{CNN}(f^{INPL}), \quad f^R = \text{CNN}(f^{INPR}) \quad (13)$$

The extracted features are then concatenated, and positional embeddings are added to form a combined feature ($f^C \in \mathbb{R}^{2\times 128\times \frac{m}{8}\times \frac{n}{8}}$). After concatenation, the concatenated features are reshaped into patches ($f_i^p \in \mathbb{R}^{p\times 196\times 128}$) for better local feature extraction. Here, p refers to the number of patches and 196 to the patch size of 14x14. $f^C \in \mathbb{R}^{2\times 128\times \frac{m}{8}\times \frac{n}{8}}$.

$$f^L = f^{INPL} + P, \quad f^R = f^{INPR} + P \quad (14)$$

$$f^C = \text{Concat}(f^L, f^R) \quad (15)$$

$$f_i^p = \text{Patch}(f^C) \quad (16)$$

The patches are then passed into the DenVisCoM block with f_i^p as input (refer to Figure 1). f_i^p is reshaped to $\mathbb{R}^{p\times 128\times 196}$ and split along the embedding dimension into $f_i^m, f_i^s \in \mathbb{R}^{p\times 64\times 196}$.

Then, f_i^s is further split along the patch dimension into the left and right embeddings ($f_{iL}^s, f_{iR}^s \in \mathbb{R}^{p/2\times 64\times 196}$). f_i^m is split into left and right patches and concatenated along the embedding dimension. This fusion along the embedding dimension allows the corresponding patches of the left and right images to be passed through the SSM branch simultaneously for joint learning, while f_{iL}^s and f_{iR}^s pass through the each symmetric convolution branches.

After passing through the symmetric branches, f_{iL}^s and f_{iR}^s are concatenated along the patch dimension back into f_i^s. The output of the SSM branch ($f_i^m \in \mathbb{R}^{p/2\times 128\times 196}$) is unfused to the original tensor $f_i^m \in \mathbb{R}^{p/2\times 128\times 196}$ for further propagation in the network. Finally, f_i^m and f_i^s are concatenated along the embedding dimension and reshaped into f_i^p and passed through a linear projection layer to get the output of the DenVisCoM as f_{iout}^p.

To further enhance joint learning between the pair of input images, we include both self-attention and cross-attention mechanisms. Multi-head attention improves the diversity of attention heads, allowing the model to capture multiple aspects of the feature space. To reduce computational complexity, we scale the number of attention heads across stages while maintaining a constant number of parameters. Each attention block includes two components: self-attention within each sequence and cross-attention between corresponding image sequences. The self-attention is applied to individual feature maps, followed by cross-attention to enable joint learning between the image pair.

Inside the attention block, f_{iout}^p is split into $f_L^a, f_R^a \in \mathbb{R}^{p/2\times 196\times 128}$. During self-attention, f_L^a and f_R^a are the key, query, and value for the left and right images, respectively. During cross-attention, first f_R^a is the query and f_L^a acts as the key and value, then f_L^a becomes the query and f_R^a acts as the key and value.

$$f_L^a = \text{SelfAttn}(f_L^a) \quad (17)$$

$$f_R^a = \text{SelfAttn}(f_R^a) \quad (18)$$

$$f_L^a = \text{CrossAttn}(Q = f_R^a, K = f_L^a, V = f_L^a) \quad (19)$$

$$f_R^a = \text{CrossAttn}(Q = f_L^a, K = f_R^a, V = f_R^a) \quad (20)$$

Thus, each image feature map attends to both itself and the other image's feature map, enabling richer multi-view representations and improving the overall model's ability to learn visual correspondences between images. After the attention block, f_L^a and f_R^a are concatenated along the patch dimension to reconstruct f_{ia}^p.

The architecture consists of two sequential Mamba and attention blocks. The Mamba and attention blocks are interleaved for improved feature representation and handling of long-range feature dependencies. The interleaving is done for n times, representing the depth with h attention heads. In the first set of Mamba and attention blocks, after the first n passes, the patch size is reduced to 7, forming a tensor $f_j^p \in \mathbb{R}^{p\times 49\times 128}$. In the first set of Mamba and attention blocks, the process is repeated for $n/2$ passes with $2h$ attention heads. Finally, f_{ja}^p is reconstructed into f^{cOut} and split along the batch dimension into f^{LOut} and f^{ROut}, which are then used for task-specific matching.

D. Task Specific Matching

For task-specific matching, we take a pair of images, which can be video frames, stereo images or posed images as I_1 and I_2. A parameter-free task-specific matching layer is used for optical flow and stereo matching. The matching layer takes the $8\times$ downsampled feature which have been processed through the Fusion Mamba, f_1,$f_2 \in \mathbb{R}^{D\times H\times W}$, here H and W denote the height and width, respectively and D refers to the feature dimension.

To compute the optical flow, global matching computes the correspondence of every location in f_1 with every location in f_2 by using matrix multiplication. To avoid very large values from the dot product,$\frac{1}{\sqrt{D}}$ is used as a normalization factor.

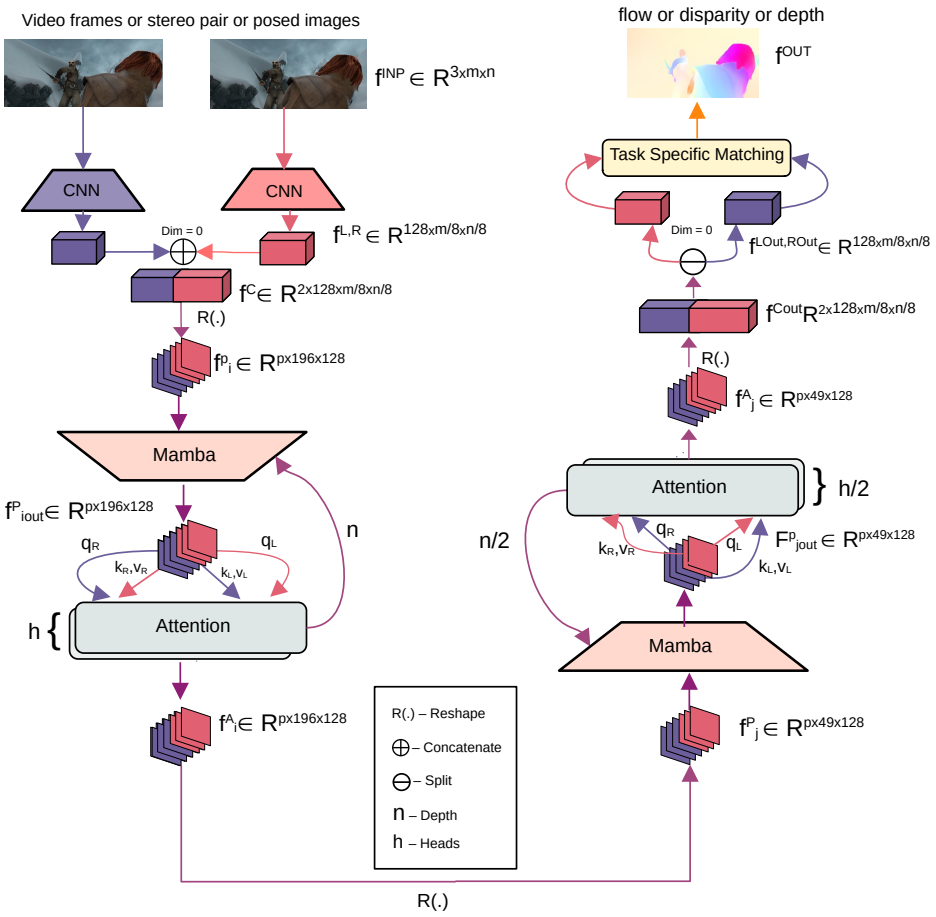

Fig. 2. Overview of the proposed hybrid model.

Further, a softmax layer is added to obtain the dense correspondences. This step normalizes the last two dimensions of C_{flow} and produces a distribution of each location in f_1 with respect to each location in f_2. The correspondence $\hat{G}_{2D}$ is obtained using the flow with 2D coordinates of pixel grid G_{2D} dimension. Finally, the optical flow V_{flow} is calculated by computing the difference between $\hat{G}_{2D}$ and G_{2D}.

The aim of stereo matching is to find disparity along the horizontal scan line. To find the 1D correspondence, it can be treated as a special case of 2D global matching. The last dimension of C_{disp} is normalized and a matching distribution. As the correspondence of each pixel in the f_1 is located to the left of the reference pixel, the upper triangle of the W × W slices of image is masked to avoid unnecessary matches. Subsequently, the 1D correspondence is obtained by computing M_{disp} with all horizontal locations $P \in \mathbb{R}^W$. The final positive disparity is obtained using the difference of corresponding coordinates in $\hat{G}_{1D}$ with 1d pixel grid G_{1D}.

IV. Experiments and Analysis

We train the flow models on the Sceneflow (Flyingthings, Monka and driving) datasets as per the MemFlow protocol[7] for 100k steps, with a batch size of 8, learning rate of 10^{-3} and AdamW optimizer. We perform zero-shot testing on KITTI15(test set) and Sintel. The primary metric used is end-point-error (EPE), the L2 distance between estimated and ground truth flow vectors. Further, EPE is also reported for motion ranges s0–10, s10–40, and s40+. We also employed the F1-all measure (F1A), which indicates the percentage of predicted flow vectors that deviate significantly from the ground truth flow, exceeding a certain threshold (usually 3 pixels) across all pixels in an image. In addition, the frame per second (FPS), memory required (M) are also reported. For disparity, a similar training protocol was followed and zero-shot testing was performed on KITTI15(test set), VKITTI1 and Sintel. We report these results in IV-A.2 for the model trained from scratch. We evaluate performance using common metrics like EPE, D1, FPS, and memory usage. EPE represents the average L1 distance between predicted and ground truth disparity, whereas D1 indicates the percentage of outliers.

We used 4x RTX A6000 (48GB) with AMD EPYC 9124 16-Core Processor for our training. For FPS and memory benchmarking we utilize a single RTX A6000.

A. Optical Flow

1) Results and Analysis: The performance of our proposed model on the KITTI dataset, as shown in Table I,

TABLE I

RESULTS ON KITTI15 FOR FLOW TASK.

Method	**EPE**	**F1-all**	$S_{0\text{-}10}$	$S_{10\text{-}40}$	S_{40+}	**FPS**	**Memory**
RAFT[4]('20)	2.45	7.9	0.43	1.18	5.7	11.7	**180.51**
Unimatch [5]('23)	2.25	7.2	0.48	1.1	5.12	33.88	236.58
MemFlow[19]('24)	3.38	12.8	0.46	1.09	5.3	35.27	241.57
HD3[20]	1.31	6.5	-	-	-	-	-
PerceiverIO[21]	4.98	5.4	-	-	-	-	-
ViMDisparity[22]	2.73	7.41	0.51	1.13	4.94	32.98	238.54
DenVisCoM (Ours)	**1.34**	**2.52**	**0.28**	**0.74**	**3.15**	**39.88**	244.89

demonstrates significant improvements over RAFT, MemFlow and Unimatch. Our model achieves the lowest End-Point Error (EPE) of 1.34 outperforming MemFlow (3.38), RAFT (2.45) and Unimatch (2.25). Additionally, it records the lowest (F1-all) at 2.52, compared to 7.9 and 7.2 for RAFT and Unimatch, respectively, indicating superior robustness in handling challenging cases of optical flow estimation. In terms of motion magnitude, our model excels particularly in the mid-range ($S_{10\text{-}40}$) and large motion categories (S_{40+}), with EPE values of 0.74 and 3.15, respectively, outperforming RAFT (1.18, 5.7) and Unimatch (1.1, 5.12). The improvement across all motion ranges underscores the effectiveness of our approach in handling complex flow estimation scenarios. We have also compared our proposed method against a non-causal Mamba implementation VSSD[16] and MambaVision[9](see IV), and our proposed method significantly outperforms both. Overall, the proposed architecture consistently demonstrates superior accuracy, particularly for mid-range and large motion estimation on KITTI, and shows competitive performance. The combination of cross-attention and proposed Mamba blocks significantly enhances the model's ability to handle diverse motion magnitudes, although further optimization may be necessary for large displacements in more complex datasets.

The proposed model shows a significantly improved FPS compared to others, with memory requirements comparable to RAFT and Unimatch. This demonstrates that the proposed method is more effective than quadratic attention for long video sequences while maintaining accuracy.

The performance of the flow task of proposed model was evaluated on the Sintel dataset, with results indicating significant improvements over existing methods such as Unimatch and RAFT, as detailed in Table II. Specifically, our model achieved the lowest unmatched error on the Sintel (Final) dataset, with a score of 10.670, outperforming Unimatch (12.74) and FlowFormer (11.37). For the Sintel (Clean) dataset, our model ranked third in both matched and unmatched error, achieving 0.44 and 7.903, respectively, compared to Unimatch, which recorded 0.34 for matched error and 6.68 for unmatched error. Furthermore, our model also secured third place in matched error on the Sintel (Final) dataset.

These results highlight the effectiveness of proposed approach, demonstrating its ability to deliver competitive optical flow estimates with relatively minimal training effort.

TABLE II

RESULT OF FLOW TASK ON SINTEL. † REPRESENTS THE METHOD THAT USES THE LAST FRAME'S FLOW PREDICTION AS INITIALIZATION FOR SUBSEQUENT REFINEMENT, WHILE OTHER METHODS ALL USE TWO FRAMES ONLY

Methods	Sintel Clean		Sintel Final	
	matched	unmatched	matched	unmatched
FlowNet2	1.56	25.4	2.75	30.11
PWC-Net+	1.41	20.12	2.25	23.7
HD3	1.62	30.63	2.17	24.99
VCN	1.11	16.68	2.22	22.24
DICL	0.97	16.24	1.66	19.44
RAFT†[4]	0.62	9.65	1.41	14.68
GMA†	0.58	7.96	1.24	12.5
DIP†	0.52	8.92	1.28	15.49
AGFlow†	0.56	8.54	1.22	12.64
CRAFT†	0.61	8.2	1.16	12.64
FlowFormer	0.41	7.63	**0.99**	11.37
GMFlowNet	0.52	8.49	1.27	13.88
GMFlow[23]	0.65	10.56	1.32	15.8
Unimatch[5]	**0.34**	**6.68**	1.1	12.74
Proposed	0.44	7.903	1.173	**10.67**

2) Ablation on Optical Flow: The ablation study evaluates the performance of different mamba implementations like Mamba2 and VSSD against our proposed model on the KITTI15, Sintel and Flying-Chairs datasets for optical flow. The study considers two key factors: mamba block implementation and the effect of fusion on the performance of the proposed model. (See Table III). The study considers 4 architectural configurations, proposed where the corresponding left and right patches are being passed through the Mamba1 block simultaneously. The second architecture is a modification of our proposed architecture, here, the patches are passed sequentially. The third architecture replaces our implementation with Mamba2, and the fourth architecture replaces our implementation with VSSD.

KITTI15 Results: The proposed configuration demonstrates the best performance with an End-Point Error (EPE) of 0.87 and a relatively low error in handling large motions (s_{40+}: 2.24). This showcases the effectiveness of the simultaneous passing of corresponding patches through Mamba1 in capturing motion details, especially in complex flow scenarios. Both proposed without fusion and Mamba2 show competitive performance, with proposed without fusion achieving the lowest large-motion error (s_{40+}: 2.13). However, it still shows a higher EPE of 0.93 compared to proposed. The model used here was first trained on Flying-chairs for 50k epochs with a batch size of 4 and subsequently finetuned on

Kitti15(train set) for 20k steps with batchsize of 4.

Sintel Results: For the Sintel dataset, our proposed model achieves the best results in terms of overall EPE (1.92). Proposed model without fusion and Mamba2 also show good results with an EPE of 2.09 and 2.14 respectively. VSSD performs slightly worse at an EPE of 2.68. The model used here was first trained on Flying-chairs for 50k epochs with a batch size of 4 and subsequently finetuned on Sintel for 20k steps with batchsize of 4.

The ablation study reveals that the proposed model with simultaneous passing of left and right patches through Mamba1 block achieves the best performance across datasets, particularly excelling in small and large motion categories. We note that while Mamba2 performs well on KITTI15 and Sintel.

TABLE III

ABLATION STUDY FOR FLOW TASK ON KITTI15 AND SINTEL[24].

Methods	KITTI15					Sintel
	EPE	F1-all	S_{0-10}	S_{10-40}	S_{40+}	EPE
Proposed	**0.87**	**2.88**	**0.17**	0.48	2.24	**1.92**
Proposed w/o Fusion	0.93	3.05	**0.17**	**0.47**	**2.13**	2.09
Mamba2	1.07	3.42	0.19	0.52	2.49	2.14
VSSD[16]	1.74	5.74	0.33	0.85	3.89	2.68

TABLE IV

ABLATION STUDY FOR DISPARITY TASK.

		Proposed	Proposed w/o Self Attn	Proposed w/o Cross Att.	Proposed w/o Attention
Kitti15[24]	EPE	**0.27**	0.3	0.32	0.39
	D1	**0.005**	0.006	0.007	0.007
Vkitti[25]	EPE	**0.18**	0.24	0.26	0.31
	D1	**0.044**	0.045	0.047	0.047
Sintel[26]	EPE	**0.51**	0.79	0.857	0.901
	D1	0.096	**0.092**	0.092	0.094

In Figure 3, the optical flow on the Sintel dataset demonstrates the model's ability to capture fine motion between frames with high precision, as evidenced by the smooth and well-defined flow fields.

B. Disparity task

1) Results and Analysis: The proposed model demonstrates clear improvements in disparity estimation across multiple datasets, particularly in reducing outliers and enhancing efficiency (see Table IV-A.2).

On the KITTI15 dataset, it achieves an EPE of 0.27 and D1 of 0.005, significantly outperforming RAFT and Unimatch in outlier reduction. While IGEV attains a similar EPE, its higher D1 (0.37) underscores the importance of balancing accuracy with consistency, which the proposed model handles effectively. On VKITTI, the model records an EPE of 0.18 and D1 of 0.044, again outperforming RAFT and IGEV in outlier percentage, highlighting its robustness in handling complex synthetic datasets. The large error reduction achieved through cross-attention mechanisms further validates its reliability in difficult scenarios. Similarly, on the

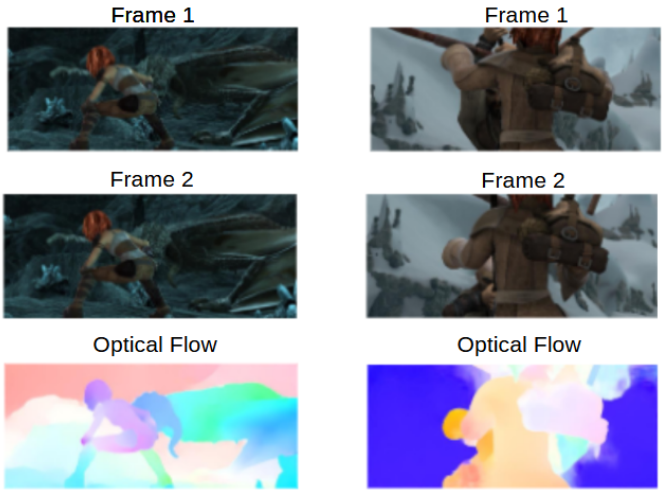

Fig. 3. Visual representation of Optical Flow task on Sintel on two samples(left column and right column).

Sintel dataset, the model performs comparably to better to Unimatch with an EPE of 0.51 and D1 of 0.096, maintaining a strong balance between accuracy and outlier minimization, even in dynamic scenes. In terms of efficiency, the proposed model delivers superior performance compared to RAFT, Anynet, and IGEV is slightly better than Unimatch (FPS of 43.6), achieving 46.03 FPS, while maintaining a competitive memory footprint of 324.85 MB, considering the significant improvements in EPE and D1. In comparison to VSSD, Mamba2, Mamba2 w/o attention and proposed w/o fusion, VisionMamba and MambaVision in the Kitti15 and Vkitti2 datasets our method outperforms them all. Moreover, the results from Table IV-A.2 clearly demonstrate that the model is highly suitable for real-time applications, combining speed and scalability with robust accuracy.

Finally, Figure 4 presents the disparity maps generated from the Vkitti2 and Sintel datasets. The vivid disparity maps reveal our model's robustness in estimating pixel-wise disparities in challenging scenes, contributing to high-quality 3D scene reconstruction.

2) Ablation on disparity task: The ablation study in Table IV highlights the performance variations between different attention mechanisms (cross-attention, self-attention, a combination of both and no attention) across three disparity datasets: KITTI15, VKITTI, and Sintel. Overall, the combination of self and cross-attention consistently achieves better accuracy, particularly in the VKitti2 dataset, where it outperforms other configurations in reducing End-Point Error (EPE) and D1. On Kitti15, the combined self and cross-attention show slightly better performance in EPE and maintain relatively low D1 errors, highlighting the robustness of the cross-attention combined with self-attention. However, on the Sintel dataset, all configurations exhibit higher EPEs, indicating that the proposed architecture may require further refinement for complex, dynamic environments. The results confirm that cross-attention combined with self-attention provides a well-balanced solution for disparity estimation,

TABLE V

DISPARITY RESULTS ACROSS DATASETS (KITTI15, VKITTI, SINTEL). D1 REPORTED BETWEEN 0-1

Method	**Kitti15**[24]		**Vkitti2**[25]		**Sintel**[26]			
	EPE	**D1**	**EPE**	**D1**	**EPE**	**D1**	**FPS**	**Memory**
Unimatch [5]('23)	1.21	0.05	1.95	0.13	1.45	**0.04**	43.6	231.45
IGEV [27]('23)	0.28	0.03	0.92	0.06	**0.32**	0.12	1.85	119.18
RAFT [4]('20)	1.08	0.05	0.92	0.06	0.45	0.13	2.82	**102.41**
Anynet [28]('18)	10.94	1.00	88.55	0.99	88.04	0.99	36	240.71
Vision Mamba [8]('24)	1.38	0.07	1.14	0.06	11.53	0.24	52.53	334
Mamba Vision [9]('24)	0.71	0.02	1.04	0.06	1.43	0.06	52.26	224.85
Mamba2('24)	0.38	0.013	0.398	0.045	0.57	0.094	44.5	309.11
Mamba2 w/o attention('24)	0.30	0.016	0.31	0.044	0.35	0.095	53.46	286.71
VSSD('24)	0.33	0.018	0.41	0.048	0.95	0.096	33.33	281.52
Proposed w/o Fusion	0.28	0.012	0.281	**0.043**	0.71	0.099	**56.59**	297.29
Proposed	**0.27**	**0.005**	**0.18**	0.044	0.51	0.096	46.03	324.85

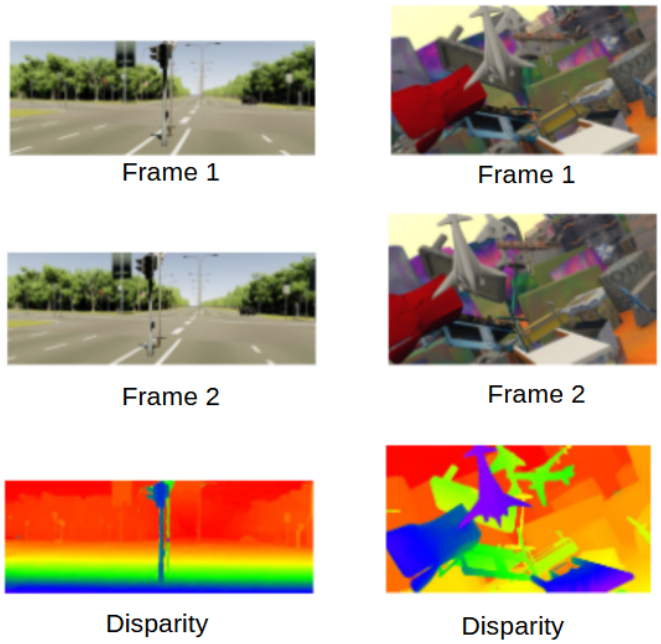

Fig. 4. Visual representation of Disparity task on Vkitti2 and Sintel.

minimizing both error and outliers across diverse datasets.

C. Flow to Disparity

The results of transferring pre-trained weights from optical flow tasks to disparity estimation demonstrate compelling findings, as outlined in Table VI. Specifically, when training the disparity estimation model using pre-trained weights derived from an optical flow task, we observe superior performance compared to training from scratch on the KITTI15 dataset. The model initialized with pre-trained weights achieves an End-Point Error (EPE) of 0.31 and a D1 score of 0.005, outperforming the model trained from scratch, which reaches an EPE of 0.39 and a D1 score of 0.006. This indicates that initializing disparity estimation training with optical flow-pretrained weights leads to a beneficial effect, yielding better accuracy metrics on this dataset. This also proves that the proposed model can produce a unified model for the dense perception task of flow and disparity.

TABLE VI

RESULT ON CROSS TASK TRANSFER FROM FLOW TO DISPARITY ON KITTI15.

Method	EPE	D1
Flow to Disparity on Proposed	**0.31**	**0.005**
Proposed from scratch	0.39	0.006

V. CONCLUSIONS

This work introduces a novel Mamba block DenVisCoM, as well as a novel unified hybrid architecture for accurate and real-time estimation of optical flow and disparity estimation. To nurture the dense correspondence of the tasks, the features from the pair of input images are fused and passed through the sequence transformation pipeline *ie* the Scan branch of the Mamba block, which consists of the SSM component. The proposed hybrid architecture uses the DenVisCoM and Transformer-based self- and cross-attention block for unified motion and 3D dense perception tasks. To conclude, experimental results and analysis showcase that the proposed model was able to tackle speed, accuracy, and memory gaps better than state-of-the-art techniques.

REFERENCES

[1] R. A. Hamzah, H. Ibrahim *et al.*, "Literature survey on stereo vision disparity map algorithms," *Journal of Sensors*, vol. 2016, 2016.

[2] Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou, and J. Zhang, "Learning for disparity estimation through feature constancy," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2811–2820.

[3] M. S. Hamid, N. Abd Manap, R. A. Hamzah, and A. F. Kadmin, "Stereo matching algorithm based on deep learning: A survey," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 5, pp. 1663–1673, 2022.

[4] L. Lipson, Z. Teed, and J. Deng, "Raft-stereo: Multilevel recurrent field transforms for stereo matching," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 218–227.

[5] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, F. Yu, D. Tao, and A. Geiger, "Unifying flow, stereo and depth estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[6] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[7] T. Dao and A. Gu, "Transformers are ssms: Generalized models and efficient algorithms through structured state space duality," 2024. [Online]. Available: https://arxiv.org/abs/2405.21060

[8] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," *arXiv preprint arXiv:2401.09417*, 2024.

[9] A. Hatamizadeh and J. Kautz, "Mambavision: A hybrid mamba-transformer vision backbone," 2024. [Online]. Available: https://arxiv.org/abs/2407.08083

[10] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, and Y. Liu, "Vmamba: Visual state space model," 2024. [Online]. Available: https://arxiv.org/abs/2401.10166

[11] Z. Li, H. Pan, K. Zhang, Y. Wang, and F. Yu, "Mambadfuse: A mamba-based dual-phase model for multi-modality image fusion," 2024. [Online]. Available: https://arxiv.org/abs/2404.08406

[12] H. Li, Q. Hu, Y. Yao, K. Yang, and P. Chen, "Cfmw: Cross-modality fusion mamba for multispectral object detection under adverse weather conditions," 2024. [Online]. Available: https://arxiv.org/abs/2404.16302

[13] W. Dong, H. Zhu, S. Lin, X. Luo, Y. Shen, X. Liu, J. Zhang, G. Guo, and B. Zhang, "Fusion-mamba for cross-modality object detection," 2024. [Online]. Available: https://arxiv.org/abs/2404.09146

[14] M. Bora, T. Anand, S. Atreya, A. Mukherjee, and A. Das, "Vim-disparity: Bridging the gap of speed, accuracy and memory for disparity map generation," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[15] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," 2024. [Online]. Available: https://arxiv.org/abs/2312.00752

[16] Y. Shi, M. Dong, M. Li, and C. Xu, "Vssd: Vision mamba with non-causal state space duality," 2024. [Online]. Available: https://arxiv.org/abs/2407.18559

[17] M. Zhou, T. Li, C. Qiao, D. Xie, G. Wang, N. Ruan, L. Mei, and Y. Yang, "Dmm: Disparity-guided multispectral mamba for oriented object detection in remote sensing," *arXiv preprint arXiv:2407.08132*, 2024.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[19] Q. Dong and Y. Fu, "Memflow: Optical flow estimation and prediction with memory," 2024. [Online]. Available: https://arxiv.org/abs/2404.04808

[20] F. Y. Zhichao Yin, Trevor Darrell, "Wscg '2004: Short communications: The 12th international conference in central europe on computer graphics, visualization and computer vision," in *Proceedings of WSCG 2004*, Plzeň, Czech Republic, Feb. 2004, pp. 275–282.

[21] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, O. Hénaff, M. M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver io: A general architecture for structured inputs & outputs," 2022. [Online]. Available: https://arxiv.org/abs/2107.14795

[22] M. Bora, T. Anand, S. Atreya, A. Mukherjee, and A. Das, "Vim-disparity: Bridging the gap of speed, accuracy and memory for disparity map generation," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[23] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao, "Gmflow: Learning optical flow via global matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8121–8130.

[24] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[25] Y. Cabon, N. Murray, and M. Humenberger, "Virtual kitti 2," *arXiv preprint arXiv:2001.10773*, 2020.

[26] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*. Springer, 2012, pp. 611–625.

[27] G. Xu, X. Wang, X. Ding, and X. Yang, "Iterative geometry encoding volume for stereo matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 919–21 928.

[28] S. Chen, D. Ergu, B. Ma, Y. Cai, and F. Liu, "Improvement of anynet-based end-to-end phased binocular stereo matching network," *Procedia Computer Science*, vol. 199, pp. 1450–1457, 2022.