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

DensePercept-NCSSD: Vision Mamba towards Real-time Dense Visual Perception with Non-Causal State Space Duality

Tushar Anand, Advik Sinha, and Abhijit Das

Abstract— **In this work, we propose an accurate and real-time optical flow and disparity estimation model by fusing pairwise input images in the proposed non-causal selective state space for dense perception tasks. We propose a non-causal Mamba block-based model that is fast and efficient and aptly manages the constraints present in a real-time applications. Our proposed model reduces inference times while maintaining high accuracy and low GPU usage for optical flow and disparity map generation. The results and analysis, and validation in real-life scenario justify that our proposed model can be used for unified real-time and accurate 3D dense perception estimation tasks. The code, along with the models, can be found at https://github.com/vimstereo/DensePerceptNCSSD**

I. INTRODUCTION

Efficient optical flow and stereo-disparity estimation from dense perception imagery is a significant challenge in several computer vision robotic systems [1]. Recent research has shifted towards deep learning-based methods, initially the CNN-based approach [2], [3], followed by transformer-based modeling [4]. Deep learning approaches demonstrate higher accuracy for dense perception tasks [4], but they are not unsuitable for a real-time applications where fast processing is required with a high frame rate and low computational demands. Works such as [5], [6] have attempted to propose a real-time model for disparity estimation, but the accuracy of such model was poor. In this direction, to dissolve the trade-off between speed and accuracy for the dense correspondence task, [7] proposes a novel disparity estimation method based on visual Mamba with low computation overhead for disparity map generation. In addition, a performance measure that can jointly evaluate the inference speed, computation overhead and the accurateness of a disparity map generation model was proposed.

Recent recent literature of Mamba, state space models (SSM) can capture long-range dependencies and are able to benefit from parallel training. Among the current SSMs, the Mamba [8] block was proposed, which can achieve linear time feature computation while maintaining similar benchmarks as Transformers for different language modeling tasks [8]. Vision Mamba (ViM) [9] was proposed by incorporating the bidirectional SSMs and position embeddings adopting patch-based analysis to adopt Mamba for the vision task. Further, VisionMamba [10] was proposed as a hybrid architecture that consists of defining a modified Mamba block along with Transformer blocks for more accurate performance than ViM. Very recently, Dao and Gu [11] proposed a state space duality (SSD) framework that designs a new architecture Mamba-2. The core layer in Mamba-2 is a refinement of Mamba's selective SSM. Further development was carried out to design the Visual State Space Model (Vmamba) [12] which introduces a cross-scanning mechanism to mitigate the problem of one-dimensional scanning in Mamba when applied to vision applications. [13] were the first to propose a non-causal mamba block which employs a multi-scan strategy and relieves the dependencies of token contribution on previous tokens.

The encouraging results of employing Mamba for disparity estimation in ViM-Disparity [7] and recent developments in the literature related to Mamba motivates us to explore the Mamba block further for more efficient and real-time unified dense correspondence tasks. Hence, in this work, we conduct an in-depth analysis of different Mamba block and their possible adaption for the real-time and accurate model for dense perception task estimation. Moreover, ViM-Disparity [7] was a hybrid model of SSM and transformer-based attention mechanism, which is found to be an important aspect to feature dense correspondence task. Therefore, there is still a need for a model for dense correspondence task that reduce the quadratic complexity of the attention block. Hence, we introduce a lightweight state-space model which can be a suitable choice to replace the quadratic attention mechanism of the transformer block by non-causal linear-based attention, thereby maintaining the performance and real-time execution for unified vision-dense correspondence tasks adopted from VSSD [13].

VSSD is a recent advancement in computer vision that serves as an alternative to Vision Transformers [14]. Recent works in SSMs, such as state space duality (SSD) [11], establish a theoretical connection between SSMs and attention mechanisms, especially through structured semi-separable matrices. We introduce a novel Mamba block based on VSSD [13] for unified dense perception tasks of optical flow and disparity estimation. We have modified the VSSD to learn rich feature representations from individual images and, further from stereo images in parallel through the non-causal mechanism, which is similar to the self and cross-attention mechanism, respectively used in the literature [4]. Further, to capture long-range within the feature, i.e. in order to capture both large and small pixel displacements, a multi-level of correlation is needed. Hence, we passed the features from the proposed Mamba block through a pyramid-based matching technique based on Gated recurrent Unit (GRU).

The specific contributions of our work are as follows:

Machine Intelligence Group, Department of Computer Science & Information Systems, Birla Institute of Technology and Science, Pilani, Hyderabad Campus, India. Email: abhijit.ads@hyderabad.bits-pilani.ac.in

- Efficient and real-time Mamba-based architecture for dense perception task estimation of flow and disparity estimation.
- A modified Mamba block DensePercept-NCSSD based on non-causal SSD that facilitates joint learning of image pair features via a visual correspondence for dense perception task.
- We perform an extensive benchmarking analysis of state-of-the-art dense perception task estimation techniques, evaluating our model's inference speed, accuracy, and memory efficiency across multiple datasets, demonstrating its suitability for real-time and resource-constrained environments.

II. Previous Works

Traditional optical flow methods fall into energy-based [15], pixel-based [16], and feature-based [17] approaches. Deep learning techniques [18] have since surpassed them, with FlowNet [19] marking a key shift. Most modern methods like RAFT [3], use convolutional cost volume, balancing large motion detection with efficiency. Coarse-to-fine [6] and iterative refinement methods [20] aim at the real-time aspect of optical flow.

Stereo disparity estimation has advanced from CNN-based cascades [21] and 3D CNNs [22] to efficient U-Net models [23] and pyramid pooling [24]. Vision Transformers (ViT) [14] inspired attention-based methods [25], [26], optimizing cost volumes, while Self-Supervised Learning (SSL) [27] further improved accuracy. Any-Net [5] combined 2D-3D CNNs for real-time disparity, and iterative refinement [28] enhanced structural consistency.

Previous research in optical flow and dense disparity perception has typically treated these tasks independently, resulting in separate models. Unified models aim to solve multiple perception tasks with a single architecture. Perceiver IO [29] introduced a transformer-based approach for optical flow and stereo matching. HD3 [30] tackled both flow and stereo but lacked transferability. Unimatch [4] was the first to propose a unified dense perception framework with a shared backbone for all tasks.

Mamba [8] provides a state-space model (SSM) alternative to Transformers, reducing computational complexity from quadratic to linear. Hybrid architectures where SSMs are combined with attention mechanisms, such as Jamba [31] in language modelling and MambaVision [10] in vision, have demonstrated state-of-the-art performance in various task. To tackle the inherent causal nature of SSM and state space duality (SSD) [11] models, VSSD [13] proposes a non causal variant suited for vision tasks by employing multi-scan strategies and relieving the dependencies of token contribution on previous tokens.

III. Proposed Methodology

A. Preliminaries

Non Causal State Space Duality: State Space Duality is a special case of selective SSMs that can be implemented in both quadratic and linear forms. The matrix transformations of selective SSMs can be represented as follows:

$$y(t) = \sum_{i=1}^{t} \mathbf{C}_t^T \mathbf{A}_{t:i+1} \mathbf{B}_i x_i, \text{where } \mathbf{A}_{t:i} = \prod_{i=2}^{t} \mathbf{A}_i \tag{1}$$

$$y = \text{SSM}(\mathbf{A}, \mathbf{B}, \mathbf{C})(x) = \mathbf{F}x, \text{where } \mathbf{F}_{ji} = \mathbf{C}_j^T \mathbf{A}_{j:i} \mathbf{B}_i \tag{2}$$

Mamba2 [11] recently simplified the matrix A into a scalar. When $\mathrm{A_i}$ is reduced to a scalar, the linear formula is as follows:

$$h(t) = \mathbf{A}_t h(t-1) + \mathbf{B}_t x(t), \quad y(t) = \mathbf{C}_t h(t). \tag{3}$$

and the quadratic form becomes:

$$y = \mathbf{F}x = \mathbf{M} \cdot \left(\mathbf{C}^T \mathbf{B}\right) x, \tag{4}$$

$$\text{where } M_{ij} = \begin{cases} A_{i+1} \times \cdots \times A_j & i > j \\ 1 & i = j \\ 0 & i < j \end{cases} \tag{5}$$

The scalar A_t adjusts the impact of the previous hidden state $h(t-1)$ and the information at the current time step. In other words, the current hidden state $h(t)$ can be seen as a linear combination of the previous hidden state and the current input, weighted by A_t and 1, respectively. Thus, if we ignore the absolute values of these two terms and focus only on their relative weighting, we can rewrite it as:

$$h(t) = h(t-1) + \frac{1}{A_t} \mathbf{B}_t x(t) = \sum_{i=1}^{t} \frac{1}{A_i} \mathbf{B}_i x(i). \tag{6}$$

In this scenario, the contribution of a specific token to the current hidden state is determined directly by $\frac{1}{A_i}$, rather than by the cumulative product of multiple coefficients. To facilitate the acquisition of global information and better accommodate non-causal image data, VSSD [13] starts with a bidirectional scanning approach. VSSD has shown that combining the results from both forward and reverse scanning can be effective for this purpose:

$$\mathbf{H}_i = \sum_{j=1}^{i} \frac{1}{A_j} \mathbf{Z}_j + \sum_{j=-L}^{-i} \frac{1}{A_{-j}} \mathbf{Z}_{-j} = \sum_{j=1}^{L} \frac{1}{A_j} \mathbf{Z}_j + \frac{1}{A_i} \mathbf{Z}_i,$$
$$\text{where } \mathbf{Z}_j = \mathbf{B}_j x(j). \tag{7}$$

VSSD [13] treats $\frac{1}{A_i} \mathbf{Z}_i$ in this equation as a bias and omits it, the above equation simplifies, resulting in all tokens sharing the same hidden state $\mathbf{H} = \sum_{j=1}^{L} \frac{1}{A_j} \mathbf{Z}_j$. In this case, the forward and reverse scanning results can be seamlessly combined to establish a global context, effectively removing the causal mask and transitioning to a non-causal format. Although these results are derived from a bidirectional scanning approach, in this non-causal format, different scanning paths yield consistent results, making specific scanning routes for capturing global information unnecessary.

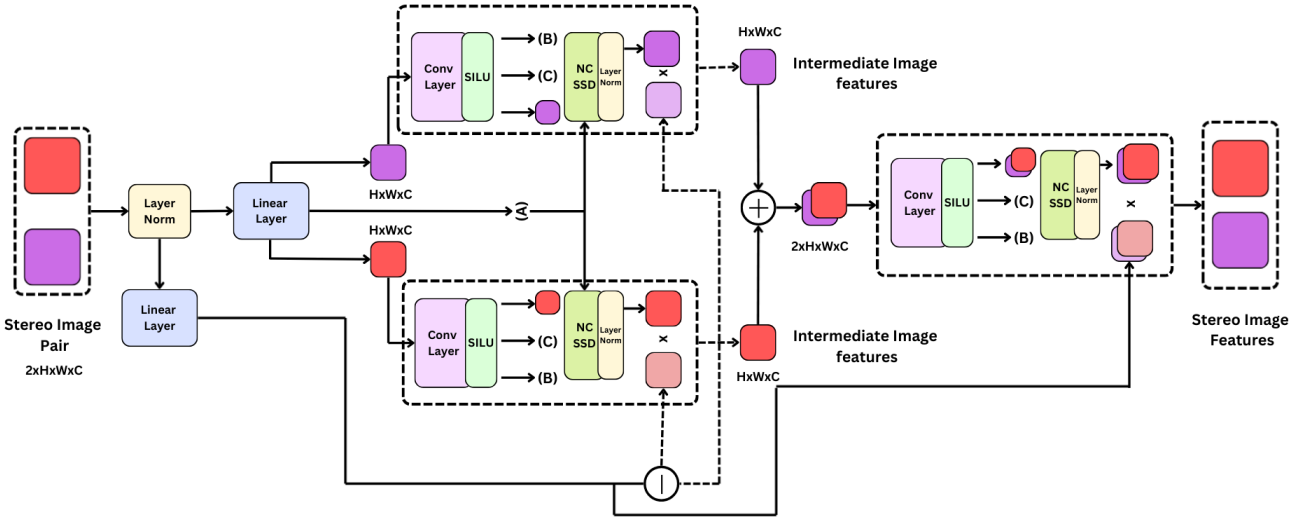

Fig. 1. Overview of the proposed DensePercept-NCSSD. The images in the stereo pair are represented in red and purple. (A),(B) and (C) are state matrices. The negative sign(-) represents a split at the batch dimension. The addition sign(+) represents concatenation at the batch dimension.

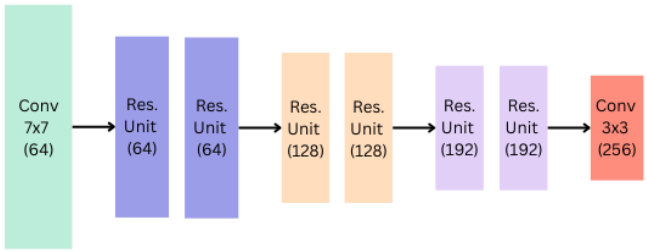

Fig. 2. Overview of the context encoder used as reference network.

Additionally, as shown in the above equation, the contribution of each token to the current hidden state is independent of its spatial distance. Consequently, transforming a flattened 2D feature map into a 1D sequence does not compromise the original structural relationships. Furthermore, the entire computation process can be performed in parallel, avoiding the recurrent methods previously needed for State Space Models (SSMs), which enhances both training and inference speeds. After revising the iteration rules for the hidden state space, VSSD updates the corresponding tensor contraction algorithm or einsum notation in linear form, in line with the Mamba2 framework [11].

$$Z = \text{contract}(\text{LD,LN} \rightarrow \text{LND})(X, B) \tag{8}$$

$$H = \text{contract}(\text{LL,LDN} \rightarrow \text{ND})(M, Z) \tag{9}$$

$$Y = \text{contract}(\text{LN,ND} \rightarrow \text{LD})(C, H) \tag{10}$$

This algorithm follows three main steps: first, it expands the input X using B; second, it unrolls the scalar SSM recurrences to form a global hidden state H; and finally, it contracts H with C. Compared to the standard SSD, while the operation in the first step remains the same, the sequence length dimension in H is removed in non-causal mode, as all tokens now share the same hidden state. In the final step, the output Y is generated by matrix multiplication of C and H. Since $M_{i,j} = \frac{1}{A_j}$, the matrix M can be simplified to a vector $m \in \mathbb{R}^L$ by removing its first dimension. Integrating m with either X or B in this setup could further simplify the transformation of the equation above.

B. Proposed Architecture

The proposed architecture can broadly be divided into two parts: 1) the feature extraction and 2) the pyramid-based marching (See Fig 3). For feature extraction, we have utilised two separate feature encoders. One set of feature encoding is done via the proposed Mamba block to generate the joint feature from the pair of image (left and right for disparity and consecutive frames for flow). The second encoder acts as the reference for matching. The first set of encoders, the proposed Mamba block DensePercept-NCSSD, obtains dense feature maps at 1/4 resolution and is jointly applied to the left and right images. A detailed description of the Mamba block is given in Figure 1. We also employ a separate context network, as shown in Figure 2, with residual blocks and downsampling layers to produce feature maps at 1/4 of the original resolution, which serve as a primary reference. Next, the matching part consists of the correlation computation that finds the visual similarity of every feature map at each pixel location and correspondence for flow/ disparity estimation at different scales at multiple iterations. Further, the relationship between the visual similarity of multi-scale is fostered by a multi-label iterative update. Now, we proceed to explain each of these components in detail.

1) Feature Extraction: The first stage of feature extraction is done in the proposed DensePercept-NCSSD (See Fig 1); where the left and right concatenated images are passed through a Layer Normalization layer to improve the stability and efficiency of the features in the pair of images. It is then split into two branches: the first branch passes through a linear layer to obtain the state transition matrix(A) and the

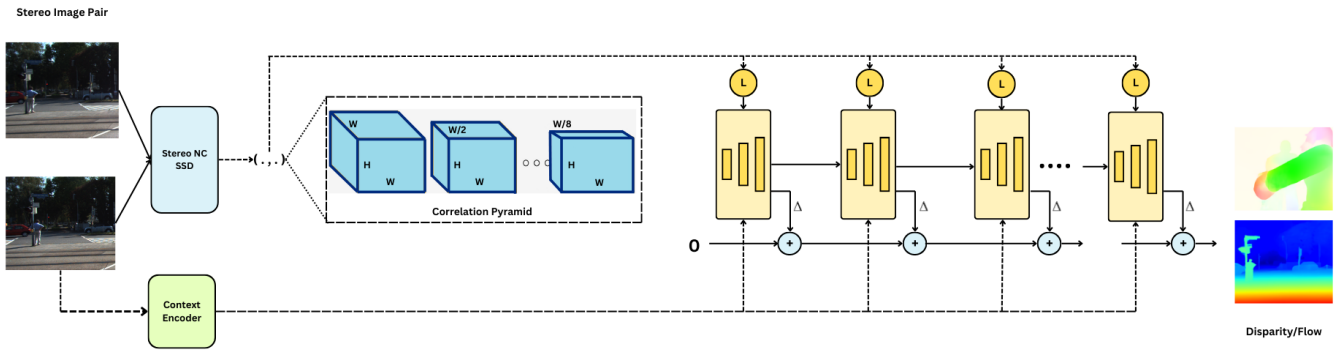

Fig. 3. Overview of the macro architecture, which consists of the feature extraction and the machining block.

left and right image patches, and the other branch passes through a linear layer to yield linear projections(Z). The individual patches from the first branch are then passed through a dedicated convolution layer and SiLU. From the SiLU, we obtain three outputs: the control input matrix (B) and the output matrix (C) by passing the activated features through a linear layer, and the other branch is simply the image patch feature (X_L/X_R). The patches, A, B, C, are then passed through the NC-SSD module and a Layer-Normalization layer to yield (Y_L/Y_R).

The linear projections are split at the batch dimension (Z_L/Z_R), and the cross product is computed. The computed cross product acts as a similar functionality of cross-attention in a transformer block, maintaining linear complexity. Further, the cross products are concatenated in a batch dimension. The concatenated tensor is subsequently used to in a similar fashion as before to obtain A,B,C and passed through the NC-SSD in a bidirectional manner. The output is finally passed through a linear layer to obtain individual left and right images for disparity or features from consecutive frames in flow. The set of feature extraction is done in the context encoder (See Fig 2), where the reference frame/ image is passed to the context encoder, which contains a CNN with six residual blocks and downsamples each image to 1/8 resolution with D feature maps.

2) Pyramid Structure-based Matching: The first step in the matching pipeline is to find the visual correspondence. Hence, we construct a correlation volume of features extracted from the left and right components of the proposed Mamba block, i.e DensePercept-NCSSD.

For optical flow, to compute visual correspondence we compute the similarity between all pairs of pixels from the left($I_l \in \mathbf{R}^{H\mathbf{x}W\mathbf{x}D}$) and right($I_r \in \mathbf{R}^{H\mathbf{x}W\mathbf{x}D}$) image. For this purpose, the correlation volume C^o_{ijkl} is computed using a dot product operation between the feature vectors.

$$C^o_{ijkl} = \sum (I_l)_{ijh} \cdot (I_r)_{klh}, \quad C^o_{ijkl} \in \mathbf{R}^{H\mathbf{x}W\mathbf{x}H\mathbf{x}W} \tag{11}$$

For disparity the 3D correlation volume computes visual similarity between pixels sharing the same y-coordinate, building a 3D correlation volume.

$$C^{disp}_{ijk} = \sum_h f_{ijh} \cdot g_{ikh}, \quad C^{disp}_i jk \in \mathbb{R}^{H\times W\times W} \tag{12}$$

Further, we use the correlation volumes obtained to generate a 4-level pyramid of correlation volumes, i.e the correlation pyramid by average pooling the last 2 dimensions of the optical flow volume and the last dimension of the disparity volume. For the the $\mathbf{k}^{th}$ step we obtain the following optical flow($\mathbf{C}^o_k$) and disparity($\mathbf{C}^{disp}_k$) correlation volumes:

$$C^o_k \in \mathbf{R}^{H\mathbf{x}W\mathbf{x}H/2^k\mathbf{x}W/2^k}, \quad C^{disp}_k \in \mathbf{R}^{H\mathbf{x}W\mathbf{x}W/2^k} \tag{13}$$

Next a lookup operator L_C, is employed to retrieve features from the correlation pyramid to find the relation between small and large pixel displacement. For disparity, it constructs a 1D grid of integer offsets around the current estimate, this grid is used to index pixels from multiple scales, which are linearly interpolated and then concatenated into a single feature map.

For optical flow, L_C maps of each pixel in the first image to its estimated correspondence in the second, defining a local neighbourhood around this point using L1 distance. This neighbourhood, spanning multiple pyramid levels, is indexed using bilinear sampling, with lower levels capturing a broader spatial context. In both cases, the retrieved values from all levels are combined into a unified feature representation.

Both stereo disparity and optical flow estimation iteratively refine predictions at each level of the pyramid, starting from an initial estimate of zero. At each iteration, the current flow or disparity estimate is used to index the generated correlation pyramid, retrieving correlation features, which are then processed by two convolutional layers. The features are then concatenated with context features and passed onto the recurrent update operator employing GRU, which refines the hidden state and predicts the next update.

For disparity estimation, a multi-resolution update strategy is employed, performing updates at 1/8, 1/16, and 1/32 of the input resolution. This allows a larger receptive field, allowing the model to capture large disparities at the lower resolutions while learning intricate features and small disparities at higher resolutions. The final disparity update and correlation lookup are performed at the highest resolution. Optical flow follows a similar iterative process, retrieving correlation features from the pyramid and processing them before updating the flow estimate.

Both methods output predictions at a lower resolution (1/4 or 1/8 of the input image) and apply convex upsampling to restore full resolution. This upsampling takes a weighted combination of a 3×3 grid of coarse resolution neighbours, with weights predicted through convolutional layers. This structured approach ensures efficient and accurate refinement of disparity and optical flow estimates.

IV. Experiments and Analysis

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) Implementation details and Metrics: We train the flow models on the Sceneflow (Flyingthings, Monka and driving) datasets as per the MemFlow protocol [32] for 100k steps, with a batch size of 8, and tested on KITTI15 and Sintel. The primary metric used is end-point-error (EPE), the l_2 distance between estimated and ground truth flow vectors. Further EPE is also reported for motion ranges s_{0-10}, s_{10-40}, and s_{40+}. 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) and SOMER introduced in [7] are measured to evaluate real-time performance. The SOMER metric, which takes into account the FPS, EPE and natural log of memory (*higher the value better is the result*) to produce a unified metric that can jointly evaluate the inference speed, computation overhead and the accuratenes to measure the real-timeliness of the algorithm.

TABLE I

Results flow on KITTI15 with comparison to exiting and related works in the literature.

Method	EPE	F1A	$S_{0\text{-}10}$	$S_{10\text{-}40}$	S_{40+}	FPS	M	SOMER
RAFT [3]	2.45	7.9	0.43	1.18	5.7	11.7	**180.51**	0.91
Unimatch [4]	2.25	7.2	0.48	1.1	5.12	33.88	236.58	2.75
HD3 [33]	1.31	6.5	-	-	-	-	-	-
UnDAF [34]	-	9.56	-	-	-	-	-	-
PerceiverIO [29]	4.98	5.4	-	-	-	-	-	-
ViMDisparity [7]	2.73	7.41	0.51	1.13	4.94	32.98	238.54	2.2
MemFlow [32]	3.38	12.8	0.46	1.09	5.3	35.27	241.57	1.90
Proposed	**0.54**	**1.4**	**0.18**	**0.45**	**1.20**	**42.93**	196.20	**15.06**

2) Results and Analysis: The proposed model outperforms all the works compared from the literature on the KITTI dataset as shown in Table I. Our model achieves state-of-the-art EPE of 0.54, demonstrating significant improvement over MemFlow (3.38), RAFT (2.45) and Unimatch (2.25). Considering F1-all, the proposed model records the lowest score of 1.43, compared to 7.9 and 7.2 for RAFT and Unimatch, respectively, exhibiting improved reliability in handling challenging optical flow estimation instances. The model also demonstrates exceptional performance in handling various motion ranges, achieving EPE values of 0.18, 0.45 and 1.20 in the low-range ($S_{0\text{-}10}$), mid-range ($S_{10\text{-}40}$) and large-range categories (S_{40+}), respectively, outperforming RAFT (0.43, 1.18, 5.7) and Unimatch (0.48, 1.1, 5.12). Our proposed method also outperforms both MambaVision [10] and VSSD, a non-causal Mamba implementation [13]. The proposed architecture consistently achieves improved accuracy on KITTI while compared to [7] across all motion ranges while maintaining competitive performance across various benchmarks. While considering FPS and SOMER the proposed method has outperformed the state-of-the-art. The proposed model demonstrated a marginally higher memory requirement than RAFT but lower compared to any other methods. Further, from Table III, we can conclude that the proposed model was able to achieve better results on Sintel dataset. From the above discussion, it can be concluded that the proposed model was able to attend reliable and real-time flow estimation better than any existing works in the literature. This demonstrates that the proposed method is more effective than quadratic attention for long video sequences while maintaining accuracy and real-time execution.

For the ablation study, we compared the model with different types of Mamba models available for the vision task, such as Mamba vision (MamVis), Vision Mamba

TABLE II

Ablation on KITTI15 for flow task.

Method	EPE	F1A	$S_{0\text{-}10}$	$S_{10\text{-}40}$	S_{40+}	FPS	M	SOMER
VSSD [13]	5.581	14.14	0.207	0.95	15.58	24.76	208.41	0.83
ViM [9]	0.76	2.57	0.24	0.58	1.73	37.36	**191.09**	9.35
MamVis [10]	0.59	1.7	0.20	0.46	1.32	36.62	218.23	11.52
Proposed W/O PM	2.04	6.95	0.42	0.87	4.79	37.83	228.17	3.41
Proposed	**0.54**	**1.43**	**0.18**	**0.45**	**1.20**	**42.93**	196.20	**15.06**

TABLE III

Result of EPE for 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

Method	Sintel Clean		Sintel Final	
	matched	unmatched	matched	unmatched
FlowNet2 [35]	1.56	25.4	2.75	30.11
PWC-Net+ [36]	1.41	20.12	2.25	23.7
HD3 [30]	1.62	30.63	2.17	24.99
UnDAF [34]	3.91	-	5.08	
VCN [37]	1.11	16.68	2.22	22.24
DICL [38]	0.97	16.24	1.66	19.44
RAFT† [3]	0.62	9.65	1.41	14.68
GMA† [39]	0.58	7.96	1.24	12.5
DIP† [40]	0.52	8.92	1.28	15.49
AGFlow† [41]	0.56	8.54	1.22	12.64
ViM-Dispartity [7]	1.43	8.9	1.71	12.67
CRAFT† [42]	0.61	8.2	1.16	12.64
PERCEIVER IO [29]	1.81	-	2.42	-
FlowFormer [18]	0.41	7.63	0.99	11.37
GMFlowNet [43]	0.52	8.49	1.27	13.88
GMFlow [4]	0.65	10.56	1.32	15.8
Unimatch [4]	0.34	6.68	1.1	12.74
Proposed	**0.28**	**6.31**	**0.95**	**10.78**

TABLE IV

DISPARITY RESULTS ACROSS DATASETS (KITTI15, VKITTI, SINTEL). COMPARED WITH DIFFERENT WORKS IN THE LITERATURE.

Method	**Kitti15** [44]		**Vkitti2** [45]		**Sintel** [46]				
	EPE	**D1**	**EPE**	**D1**	**EPE**	**D1**	**FPS**	**Memory**	**SOMER**
Unimatch [4]('23)	1.21	0.05	1.95	0.13	1.45	**0.04**	48.6	231.45	7.37
IGEV [28]('23)	0.38	0.37	0.92	5.70	**0.32**	1.18	1.85	119.18	1.01
RAFT [47]('20)	1.08	4.95	0.92	6.33	0.45	1.31	2.82	**102.41**	0.56
Anynet [5]('18)	10.94	1.00	88.55	0.99	88.04	0.99	36	240.71	0.6
ViM-Disparity('24) [7]	1.21	0.05	1.45	**0.04**	1.95	0.13	51.53	345	6.41
Proposed	**0.31**	**0.015**	**0.21**	**0.04**	0.42	0.08	**51.71**	109.93	**35.36**

(ViM) and VSSD, and types of matching techniques. It can be observed from Table II combination of pyramid based matching (PM) upsampling and proposed Mamba blocks significantly enhances the model's ability to handle diverse motion magnitudes. The multi-scale nature of the pyramid structure ensures the model can capture both long range and short-range motions. Moreover, in comparison to the existing Mamba block, the proposed Mamba block was able to attend much better results. This proves the effectiveness of the proposed Mamba block. Further, it can be observed that all the components of the proposed model and Mamba block shows a significantly improved FPS compared to others, with memory requirements. Some visual examples of flow estimation on KITTI15 Dataset are in Fig 4.

B. Disparity task

1) Implementation details and Metrics: For stereo disparity, models are trained on the Sceneflow(Flyingthings, Monka and driving) datasets as per the MemFlow protocol [32] dataset for 100k steps with batch size 16 and LR of 2e-4, and tested on KITTI15, VKITTI1 and Sintel. We evaluate performance using common metrics like EPE, D1, FPS, SOMER and memory usage. EPE represents the average L1 distance between predicted and ground truth disparity, whereas D1 indicates the percentage of outliers.

2) 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).

On the KITTI15 dataset, it achieves an EPE of 0.31 and D1 of 0.015, 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 0.21 and D1 of 0.041, again outperforming RAFT and IGEV in outlier percentage, highlighting its robustness in handling complex synthetic datasets. The large error reduction achieved through the combined learning between left and right images in the Stereo NC-SSD blocks and the pyramid based iterative improvements further validate its reliability in difficult scenarios. Similarly, on the extremely challenging Sintel dataset, the model comes in second after IGEV while performing comparably better to Unimatch with an EPE of 0.42 and D1 of 0.08, 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, IGEV and is slightly better than Unimatch (FPS of 48.6), achieving 51.53 FPS, while maintaining a competitive memory footprint of 109.93 considering the significant improvements in EPE and D1. Considering FPS and SOMER, the proposed model was again best when compared to any work from the literature. Similar to the flow proposed model requires less memory, only RAFT attends a little lower memory requirement than the proposed model. To conclude, the proposed model was able to attend reliable and real-time disparity estimation better than any existing works in the literature. Some visual results of disparity estimation on KITTI15 Dataset are in Fig 5.

Ablation: In comparison to VSSD, Mamba2, VisionMamba and MambaVision in the Kitti15, Vkitti2 and Sintel datasets, our method outperformed all scenarios. This proves the effectiveness of the proposed Mamba block for the deparity task. The pyramid based matching is also found to be effective as much better accurateness was found when it is applied, although a little drop in the FPS can be observed. Moreover, the results from Table IV demonstrate that the model is highly suitable for real-time applications, combining speed and scalability with robust accuracy.

Overall observation: The proposed architecture demonstrates robust and consistent performance across multiple tasks, including optical flow and disparity estimation with different conditions (indoor, outdoor, and lighting), which proves its robust performance. By integrating the proposed Mamba block based on non-causal linear-based attention mechanisms from SSDs with traditional pyramid-based refinement mechanisms, the model achieves a significant reduction in error while exhibiting higher FPS and lower memory consumption in comparison to previous works.

V. CONCLUSION

Dense perception task estimation is a well-explored area in the robotic vision community. It is well known that accuracy and real-time generation are found to be a trade-off. Further, most existing techniques aim to enhance the accuracy of the system. This work attempted to extensively analyze the aforementioned trade-off of recent dense perception task

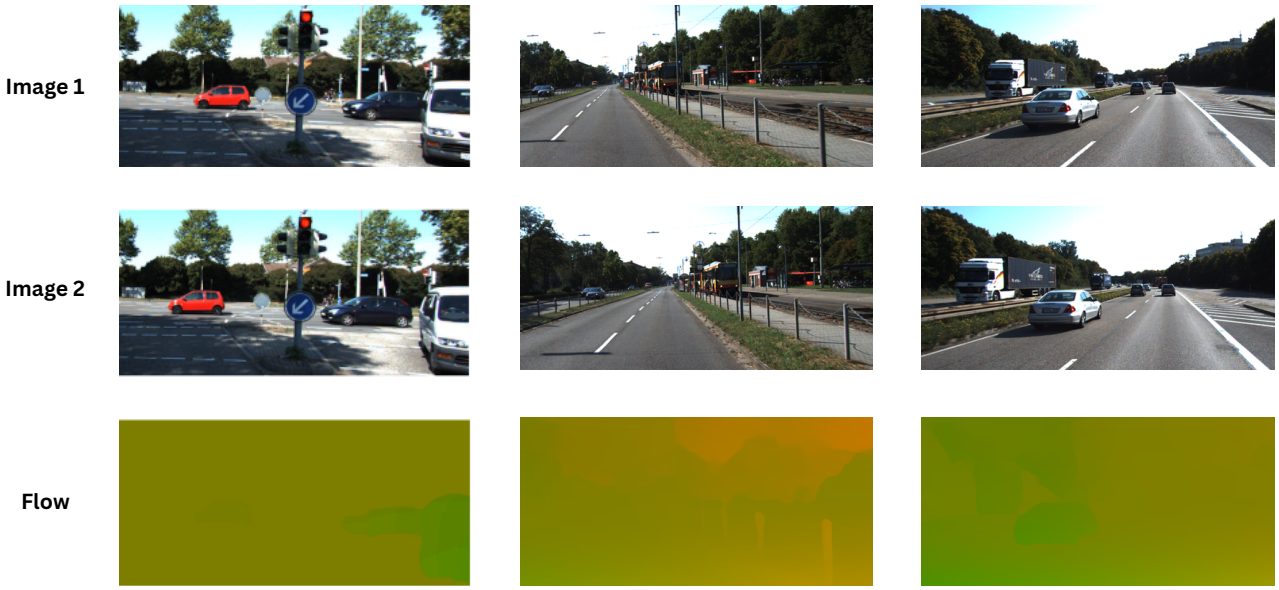

Fig. 4. Visual representation of Flow on KITTI15 Dataset

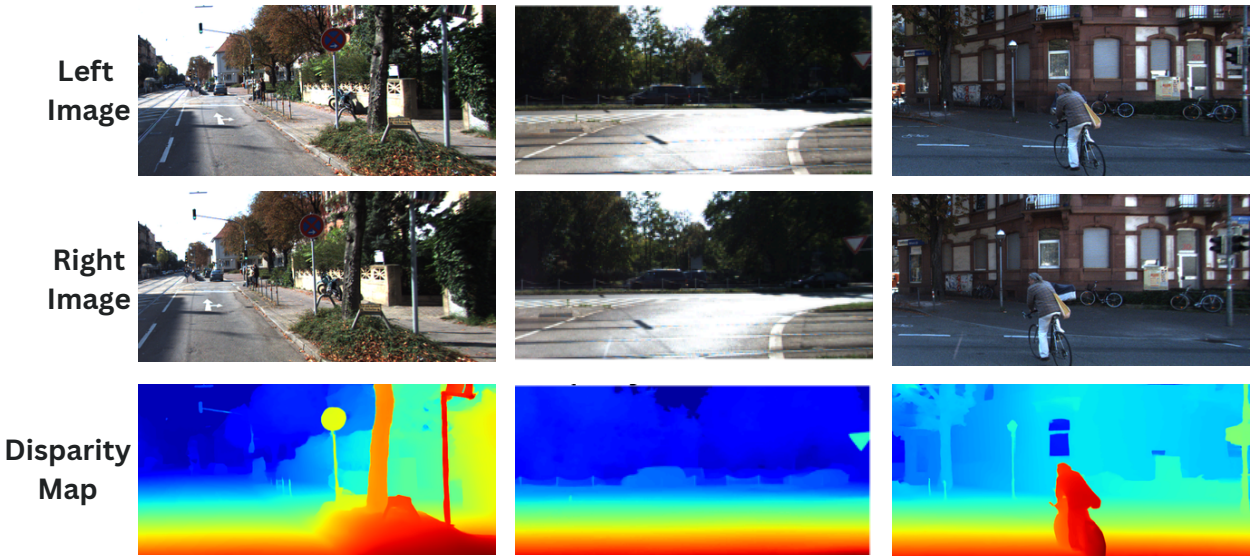

Fig. 5. Visual representation of disparity on KITTI15 Dataset

TABLE V

ABLATION ON DISPARITY TASK ACROSS DATASETS (KITTI15, VKITTI, SINTEL).

Method	**Kitti15** [44]		**Vkitti2** [45]		**Sintel** [46]				
	EPE	**D1**	**EPE**	**D1**	**EPE**	**D1**	**FPS**	**Memory**	**SOMER**
Vision Mamba [9]('24)	1.38	0.07	1.14	0.06	11.53	0.24	51.31	334	6.39
VSSD('24)	0.33	0.018	0.41	0.048	0.95	0.096	33.33	281.52	17.91
Mamba Vision [10]('24)	0.37	0.02	0.32	0.05	0.098	0.086	49.20	141.61	26.84
Mamba2('24)	0.38	0.018	0.31	0.048	0.62	0.094	50.73	101.48	28.89
Proposed W/O PM	0.71	0.02	1.04	0.06	1.43	0.06	**52.26**	**98.27**	16.04
Proposed	**0.31**	**0.015**	**0.21**	**0.041**	0.42	0.08	51.71	109.93	**35.36**

estimation methodologies on standard datasets. Concluding from this, we propose a stereo Non-causal SSD model that replaces the quadratic attention of the vision transformer block by linear attention of SSD to bridge the gap of time, memory requirement and accuracy of a real-time dense perception task estimation for flow and disparity. The results and analysis conclude that we could dissolve the speed, accuracy and memory gap with remarkable improvements.

REFERENCES

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

[2] H. et al, "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.

[3] L. et al, "Raft-stereo: Multilevel recurrent field transforms for stereo matching," in *International Conference on 3D Vision*. IEEE, 2021.

[4] X. et al., "Gmflow: Learning optical flow via global matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8121–8130.

[5] S. Chen et al., "Improvement of anynet-based end-to-end phased binocular stereo matching network," *Procedia Computer Science*, 2022.

[6] Y. Deng et al., "Detail preserving coarse-to-fine matching for stereo matching and optical flow," *IEEE Transactions on Image Processing*, vol. 30, pp. 5835–5847, 2021.

[7] 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," 2025. [Online]. Available: https://arxiv.org/abs/2412.16745

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

[9] Z. et al, "Vision mamba: Efficient visual representation learning with bidirectional state space model," *arXiv preprint arXiv:2401.09417*, 2024.

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

[11] T. D. et al., "Transformers are ssms: Generalized models and efficient algorithms through structured state space duality," 2024. [Online]. Available: https://arxiv.org/abs/2405.21060

[12] Y. L. et al., "Vmamba: Visual state space model," 2024. [Online]. Available: https://arxiv.org/abs/2401.10166

[13] Y. S. et al., "Vssd: Vision mamba with non-causal state space duality," 2024. [Online]. Available: https://arxiv.org/abs/2407.18559

[14] D. et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[15] F. Steinbrücker et al., "Large displacement optical flow computation withoutwarping," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 1609–1614.

[16] M. M. et al., "Discrete optimization for optical flow," in *German Conference on Pattern Recognition (GCPR)*, 2015.

[17] J. R. et al., "Epicflow: Edge-preserving interpolation of correspondences for optical flow," 2015. [Online]. Available: https://arxiv.org/abs/1501.02565

[18] Z. H. et al., "Flowformer: A transformer architecture for optical flow," 2022. [Online]. Available: https://arxiv.org/abs/2203.16194

[19] P. F. et al., "Flownet: Learning optical flow with convolutional networks," 2015. [Online]. Available: https://arxiv.org/abs/1504.06852

[20] M. Mühlhausen et al., "Iterative optical flow refinement for high resolution images," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 1282–1286.

[21] N. Mayer et al., "Large dataset to train cnns for disparity, optical flow, and scene flow estimation," in *Conference on computer vision and pattern recognition*, 2016.

[22] J.-R. Chang et al., "Pyramid stereo matching network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[23] W. et al., "Anytime stereo image depth estimation on mobile devices," in *ICRA)*. IEEE, 2019.

[24] Y. et al., "Hierarchical deep stereo matching on high-resolution images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5515–5524.

[25] X. et al., "Attention concatenation volume for accurate and efficient stereo matching," in *Proceedings of the conference on computer vision and pattern recognition*, 2022.

[26] Z. Shen et al., "Pcw-net: Pyramid combination and warping cost volume for stereo matching," in *European conference on computer vision*. Springer, 2022, pp. 280–297.

[27] X. et al, "Digging into uncertainty in self-supervised multi-view stereo," in *Proceedings of the International Conference on Computer Vision*, 2021.

[28] ——, "Iterative geometry encoding volume for stereo matching," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2023.

[29] A. J. et al., "Perceiver io: A general architecture for structured inputs & outputs," 2022. [Online]. Available: https://arxiv.org/abs/2107.14795

[30] Z. Yin, T. Darrell, and F. Yu, "Hierarchical discrete distribution decomposition for match density estimation," 2019. [Online]. Available: https://arxiv.org/abs/1812.06264

[31] O. L. et al., "Jamba: A hybrid transformer-mamba language model," 2024. [Online]. Available: https://arxiv.org/abs/2403.19887

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

[33] X. Jiang and Y. Ji, "Hd3: Distributed dueling dqn with discrete-continuous hybrid action spaces for live video streaming," in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2632–2636. [Online]. Available: https://doi.org/10.1145/3343031.3356052

[34] H. Wang, R. Fan, P. Cai, M. Liu, and L. Wang, "Undaf: A general unsupervised domain adaptation framework for disparity or optical flow estimation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 01–07.

[35] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," 2016. [Online]. Available: https://arxiv.org/abs/1612.01925

[36] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," 2018. [Online]. Available: https://arxiv.org/abs/1709.02371

[37] G. Yang and D. Ramanan, "Volumetric correspondence networks for optical flow," in *NeurIPS*, 2019.

[38] W. et al., "Displacement-invariant matching cost learning for accurate optical flow estimation," 2020. [Online]. Available: https://arxiv.org/abs/2010.14851

[39] S. Jiang, D. Campbell, Y. Lu, H. Li, and R. Hartley, "Learning to estimate hidden motions with global motion aggregation," 2021. [Online]. Available: https://arxiv.org/abs/2104.02409

[40] Z. Zheng, N. Nie, Z. Ling, P. Xiong, J. Liu, H. Wang, and J. Li, "Dip: Deep inverse patchmatch for high-resolution optical flow," 2022. [Online]. Available: https://arxiv.org/abs/2204.00330

[41] A. Luo, F. Yang, K. Luo, X. Li, H. Fan, and S. Liu, "Learning optical flow with adaptive graph reasoning," 2022. [Online]. Available: https://arxiv.org/abs/2202.03857

[42] X. Sui, S. Li, X. Geng, Y. Wu, X. Xu, Y. Liu, R. Goh, and H. Zhu, "Craft: Cross-attentional flow transformer for robust optical flow," 2022. [Online]. Available: https://arxiv.org/abs/2203.16896

[43] S. Zhao, L. Zhao, Z. Zhang, E. Zhou, and D. Metaxas, "Global matching with overlapping attention for optical flow estimation," 2022. [Online]. Available: https://arxiv.org/abs/2203.11335

[44] A. Geiger et al., "The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

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

[46] D. J. Butler et al., "A naturalistic open source movie for optical flow evaluation," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy*. Springer, 2012.

[47] Z. Teed et al., "Raft: Recurrent all-pairs field transforms for optical flow," in *Computer Vision–ECCV 2020*. Springer, 2020.