sciendo

JAISCR, 2026, Vol. 16, No. 2, pp. 163 – 184
10.2478/jaiscr-2026-0009

# MAMBASC: A FEATURE MATCHING METHOD USING MAMBA2 WITH SELF AND CROSS-ATTENTION FOR MULTIMODAL IMAGES

Rongrui Teng[1,3], Yun Liao[1,2,3,*], Wei Wang[1,3] Qing Duan[1,3], Junhui Liu[1,3], Fangwei Jin[1]

[1]*Yunnan University Kunming, 650506, China*

[2]*SYunnan Lanyi Network Technology Co Kunming, 650000, China*

[3]*Yunnan Key Laboratory of Software Engineering Kunming, 650506, China*

[*]*E-mail: liaoyun@ynu.edu.cn*

*Submitted: 25th June 2025; Accepted: 27th December 2025*

**Abstract**

Multimodal image matching remains a challenging yet essential task in the field of computer vision. In recent years, detector-free methods have emerged as promising approaches, achieving high matching accuracy by leveraging global modeling capabilities. While transformer-based methods are effective, they often suffer from significant computational overhead, limiting their efficiency.To address this, we propose MambaSC, a novel framework that integrates Mamba with self-attention and cross-attention mechanisms to balance accuracy and efficiency. Specifically, MambaSC introduces the M2Backbone for efficient feature extraction and the MSC Module to enhance feature interaction and alignment.Extensive experiments across multiple multimodal image datasets demonstrate that MambaSC consistently outperforms state-of-the-art methods while maintaining computational efficiency, making it a compelling solution for complex multimodal image matching scenarios. Code is available at: https://github.com/LiaoYun0x0/MambaSC.

**Keywords:** feature matching, multi-modal image, mamba, transformer, attention

## 1 Introduction

Image matching is a fundamental problem in computer vision that aims to identify semantically or structurally consistent regions across images and align them at the pixel or feature level. It underpins various downstream tasks such as 3D reconstruction, panoramic stitching, autonomous navigation, and remote sensing. Recently, multi-modal image matching has gained increasing attention for its ability to establish correspondences between images from different modalities, such as infrared–visible and SAR–optical pairs. However, large discrepancies in illumination, texture, and structure across modalities make this task far more challenging than single-modal matching. Bridging these modality gaps requires more robust and adaptive feature representations.

Traditional detector-based methods [1-6] rely on keypoint detectors and descriptors to extract local features. Although they offer strong geometric robustness, their performance heavily depends on detector quality and degrades under low-texture or cross-modal conditions. In contrast, detector-free methods [7-12] perform dense matching directly on learned feature maps, improving structural expressiveness and modality adaptability. Yet, most still struggle to balance accuracy and computational efficiency, limiting their scalability in resource-constrained environments.

To address this challenge, we propose MambaSC, a Mamba-based cross-modal image matching framework composed of a Mamba2-driven feature extraction backbone (M2Backbone) and a Mamba2-enhanced Self- and Cross-Attention module (MSC). The M2Backbone integrates hierarchical convolutions, positional encoding, and the Mamba2 module to efficiently extract discriminative features while capturing global context through fine-grained state-space modeling. Built upon these backbone features, MSC enhances representation quality by integrating Mamba2-based state-space modeling with self- and cross-attention mechanisms, jointly enabling intra-modal feature refinement and inter-modal interaction for effective cross-modal alignment. Together, M2Backbone and MSC form a unified MambaSC framework that delivers robust and efficient feature representations for complex multimodal image matching tasks, as validated by extensive experiments. As shown in Figure 1, our framework achieves high accuracy (61.24%) with low inference time (42 s), demonstrating a favorable trade-off between precision and efficiency. The main contributions are summarized as follows:

1. Mamba2 Backbone (M2Backbone): We propose a Mamba2-based backbone that efficiently captures multi-scale features. Integrating convolutional layers, positional encoding, and Mamba2 modules, it compresses spatial dimensions and enhances global context for robust image matching representations.

2. Mamba-Enhanced Self- and Cross-Attention Module (MSC): We design a Mamba2-enhanced module integrating self- and cross-attention to strengthen inter-modal interaction. This hybrid structure aligns heterogeneous features efficiently while preserving strong global dependency modeling and representation capability.

3. Extensive Experiments and Results: Comprehensive experiments on multiple multimodal datasets verify the effectiveness and versatility of our approach. The joint use of M2Backbone and MSC yields significant gains in matching accuracy and robustness, achieving superior performance across diverse cross-modal image matching scenarios.

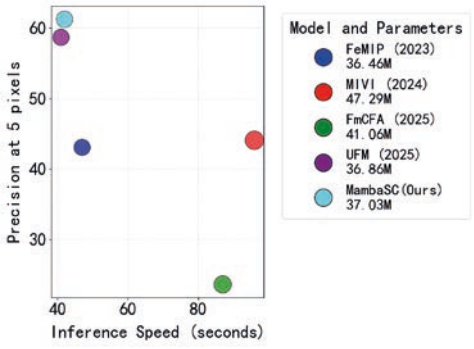


**Figure 1**. Accuracy vs. Inference Speed on NYU-DepthV2 Dataset, MambaSC outperforms most other methods with the highest accuracy, faster inference speed, and lower model size.

## 2 Related Work

### 2.1 Detector-based local feature matching method

Detector-based local feature matching methods have long formed the foundation of image matching but suffer from inherent limitations. Classical approaches such as SIFT [1], SURF [2], and ORB [3] detect keypoints invariant to scale, rotation, or illumination, enabling robust matching across viewpoints. However, their reliance on hand-crafted features often limits efficiency, adaptability, and generalization in complex scenes. Recent learning-based methods alleviate these issues by jointly learning keypoint detection and description. SuperPoint [4] adopts a self-supervised CNN framework to improve robustness, while SuperGlue [5] employs graph neural networks [13] to model contextual relationships and achieve globally consistent matching. LightGlue [6] further improves efficiency through adaptive depth control, delivering comparable accuracy at higher speeds. OmniGlue

[14] enhances cross-domain generalization via decoupled attention, and DeDoDe [15] improves local feature robustness by decoupling detection and description using 3D geometric priors. Nevertheless, these approaches remain fundamentally dependent on accurate keypoint localization, which is unreliable in low-texture regions, extreme illumination, or severe viewpoint changes.

## 2.2 Detector-free local feature matching method

Detector-free local feature matching methods eliminate the need for explicit keypoint detection, instead performing matching directly on dense, high-resolution or high-level feature maps. LoFTR [7], a pioneering method in this direction, employs self- and cross-attention mechanisms to facilitate coarse-to-fine dense matching, significantly enhancing performance in challenging conditions. Efficient LOFTR [16] further accelerates the process by up to 2.5× through adaptive token aggregation and two-stage relevance refinement, surpassing SuperPoint+LightGlue in both speed and accuracy. DISK [17] introduces a probabilistic feature learning framework trained with policy gradient methods to boost matching robustness. ASpanFormer leverages multi-scale attention to capture hierarchical visual structures, while QuadTree [18] applies quadtree-based feature partitioning to reduce computational complexity and improve efficiency. For dense correspondences, DKM [11] utilizes kernel regression to refine matching precision, and RoMa [12] integrates frozen DINOv2 features with fine-tuned convolutional features and a Transformer-based decoder for enhanced cross-domain performance. PDCNet [19] estimates dense optical flow and its uncertainty through a constrained mixture model enhanced by self-supervised training. DeepMatcher [20] employs a SlimFormer architecture for efficient global context aggregation. Despite their success, detector-free methods suffer from high computational cost due to dense similarity computation on high-resolution feature maps.

## 2.3 Multimodal image matching method

The substantial modality gap between different sensors in multimodal images poses significant challenges to traditional matching methods, motivating the development of specialized multimodal feature matching approaches. MICM [21] effectively enhances the matching performance of multimodal remote sensing images by integrating the multi-scale local feature learning ability of CNN with the attention mechanism of Transformer. LT-Former [22] addresses limited labeled data and nonlinear radiometric differences by combining a lightweight Transformer with a pyramid structure and an LT Loss triplet objective. MFAM-RegNet [23] effectively overcomes the nonlinear radiation differences and speckle noise interference between SAR and optical images by combining the point features extracted by graph neural networks and the edge features constructed based on graph topology, and using linear and quadratic contrastive learning for feature alignment. SOFT [24] improves rotational invariance by constructing second-order tensor orientation features and employs a global–local iterative optimization strategy to refine correspondences. Despite these advances, existing methods still struggle to achieve an optimal balance between matching accuracy and computational efficiency.

## 2.4 Mamba

In recent years, Mamba and related state-space models have demonstrated strong representation capability and computational efficiency in both NLP and computer vision tasks. S4 [25] addresses the efficiency and memory limitations of recurrent and attention-based models, achieving strong performance on sequence modeling benchmarks. Building on S4, S6 (Mamba) [26] improves numerical stability and hardware compatibility, enabling scalable and efficient sequence modeling on modern accelerators. Mamba2 [27] further extends the S4/S6 framework by optimizing state transitions and input–output representations, revealing connections between Mamba and Transformer architectures and enabling more efficient feature extraction. Jamba [28] combines Transformer and Mamba layers with Mixture-of-Experts, achieving state-of-the-art long-context modeling up to 256K tokens within a single 80GB GPU. Recent studies have also explored the applicability of Mamba to vision tasks. MambaOut [29] analyzes Mamba's strengths and limitations in vision, showing limited gains in classification but promising potential for long-sequence tasks such as detection and segmentation. Vim [30] replaces self-attention with bidirectional Mamba blocks and po-

sitional embeddings, demonstrating the feasibility of Mamba-based alternatives for visual representation learning.

# 3 Methods

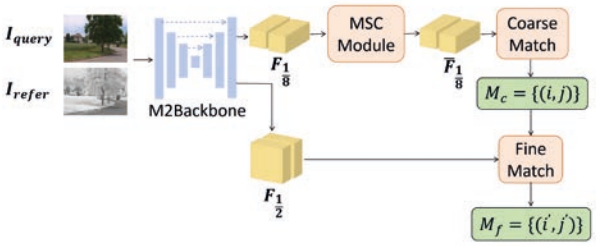


**Figure 2**. Architectural overview of MambaSC. The Mamba backbone extracts features from $I_{query}$ and $I_{refer}$, producing 1/2-scale maps for fine matching and 1/8-scale maps for enhancement. The 1/8 maps are processed by a Mamba module integrating self- and cross-attention, followed by a coarse-to-fine matching mechanism.

For the image $I_{query}$ and the image $I_{refer}$, feature extraction is performed using the Mamba-based backbone (Section 3.1). Two sets of feature maps are generated: a half-resolution (1/2 scale) map for fine matching, and a lower-resolution (1/8 scale) map for subsequent feature enhancement. The 1/8 scale feature map is further processed by the Mamba module, which integrates both self-attention and cross-attention mechanisms (Section 3.2). A hierarchical two-stage matching mechanism is utilized, starting from coarse matching and followed by fine refinement (Section 3.3). Each matching stage is guided by a specifically tailored supervision scheme to ensure effective training (Section 3.4). The overall structure of MambaSC is illustrated in Figure 2.

## 3.1 Mamba2 Backbone

The extraction block is responsible for generating feature representations at 1/2 and 1/8 of the input resolution, as illustrated in Figure 3. Within the encoder pathway, the image undergoes gradual downsampling, which compresses spatial dimensions while progressively increasing feature depth across stages (c1 = 128, c2 = 192, c3 = 256, c4 = 512). Positional encoding is iteratively added at each layer, and the data is processed through both a convolutional neural network (CNN) module and the M2 (Mamba2) module, which operate independently with separate weights. Specifically, as shown in Figure 5, in the M2 module, the input first passes through a linear projection layer, followed by a CNN block that extracts local spatial features. The output of the CNN is then activated and fed into the State Space Model (SSM) to capture long-range dependencies. The outputs from the CNN and SSM branches are combined through element-wise multiplication after activation to enhance feature interaction. The resulting features are subsequently normalized, linearly projected, and used to produce the final output. This hierarchical encoding process ultimately yields four feature maps at different scales for downstream processing. The overall operation can be expressed mathematically as follows:

$$I_{\frac{1}{2}} = Cnn(Image) + P \tag{1}$$

$$I_{\frac{1}{4}} = Cnn(I_{\frac{1}{2}}) + P \tag{2}$$

$$I_{\frac{1}{8}} = Cnn(M2(I_{\frac{1}{4}})) \tag{3}$$

$$I_{\frac{1}{16}} = Cnn(M2(I_{\frac{1}{8}})) \tag{4}$$

where $Image$ is represented as a three-dimensional tensor. $CNN(\cdot)$ denotes the convolutional neural network layer, and $P$ represents the positional encoding added at each stage, and $M2(\cdot)$ refers to the Mamba2 module. Through this sequence, four feature maps with varying resolutions are obtained: $I_{\frac{1}{2}}, I_{\frac{1}{4}}, I_{\frac{1}{8}}, I_{\frac{1}{16}}$ corresponding to 1/2, 1/4, 1/8, and 1/16 of the original image resolution, respectively. In the Decoder phase, the image is first upsampled using bilinear interpolation. The features are then summed to produce an intermediate feature map $f_{\frac{1}{8}}$. This map is subsequently convolved with four different kernels, each with distinct receptive fields. The weighted sum of the convolution results produces the final output feature map $\dot{f}_{\frac{1}{8}}$. The operations in this stage can be formally described as follows:

$$f_{\frac{1}{8}} = Up(I_{\frac{1}{16}}) + I_{\frac{1}{8}} \tag{5}$$

$$\dot{f}_{\frac{1}{8}} = Conv_{1\times1}(f_{\frac{1}{8}}) + Conv_{3\times3}(f_{\frac{1}{8}}) + Conv_{5\times5}(f_{\frac{1}{8}}) + Conv_{7\times7}(f_{\frac{1}{8}}). \tag{6}$$

In this process, $Up(\cdot)$ denotes the upsampling operation using bilinear interpolation, $f_{\frac{1}{8}}$ represents the weighted feature map, and $\dot{f}_{\frac{1}{8}}$ corresponds to

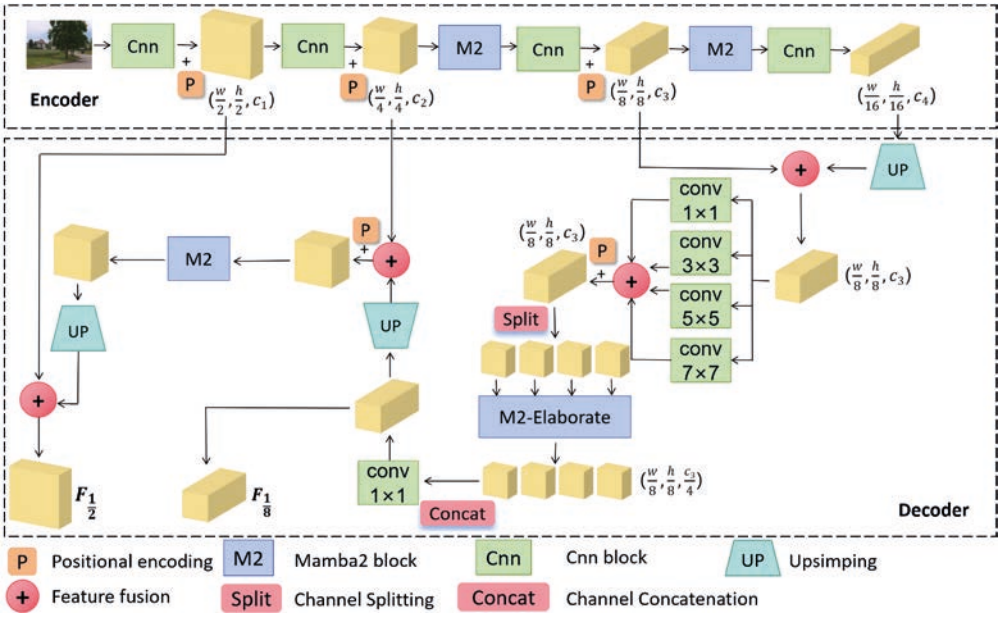


**Figure 3**. Detailed Structures of M2backbone within the MambaSC Framework. The encoder progressively downsamples the input image through CNN and M2 modules with iterative positional encoding, generating feature maps at 1/2, 1/4, 1/8, and 1/16 resolutions. In the decoder, the 1/16-scale feature is upsampled and fused with the 1/8 encoder output. The resulting feature undergoes multi-scale convolution, positional encoding, and channel-wise splitting into four parts, each processed by a shared Mamba2Block. The outputs are concatenated and merged via 1×1 convolution to form the 1/8 feature $F_{\frac{1}{8}}$. This feature is further upsampled and fused with the 1/4 and 1/2 encoder outputs, ultimately producing the final 1/2-resolution feature map $F_{\frac{1}{2}}$.

the 1/8 scale feature map, which integrates convolution kernels with different receptive fields. Next, this study adds position encoding to the feature map again, and the formula is as follows:

$$\bar{f}_{\frac{1}{8}} = \dot{f}_{\frac{1}{8}} + P. \tag{7}$$

To further refine feature processing, a finer-grained state-space model is employed, where each state-space model processes fewer dimensions. Specifically, the input feature map is first divided along the channel dimension into four parts, each of which is processed by a more refined Mamba2Block(with shared weights at this stage). The resulting features are then concatenated along the channel dimension. Finally, a 1×1 convolution kernel is applied to integrate the channels of the concatenated features, producing the output feature map. This approach enhances computational efficiency by processing smaller channel groups in parallel while maintaining feature diversity through shared weights. The 1×1 convolution then dynamically fuses these specialized features, balancing model capacity and inference speed. This design is particularly effective for high-resolution inputs where fine-grained feature extraction is critical. This process can be expressed mathematically as follows:

$$\{f_1, f_2, f_3, f_4\} = Split(\bar{f}_{\frac{1}{8}}, dim = channels, parts = 4). \tag{8}$$

This step divides the input feature map $\bar{f}_{\frac{1}{8}}$ into four equal parts along the channel dimension, denoted as $f_1$, $f_2$, $f_3$, and $f_4$. This division reduces the dimensionality that each state-space model needs to process, providing a foundation for subsequent fine-grained operations. Next, the segmented features are input into a more sophisticated Mamba2 module. The formula is as follows:

$$f_i' = M2(f_i), i \in \{1,2,3,4\}. \tag{9}$$

Each part of the feature map $f_i$ is sequentially input into the Mamba2Block for processing, yielding the enhanced feature map $f_i'$. At this stage, all Mamba2Block modules share the same set of weights, which improves feature representation capability while ensuring consistency and computational efficiency. Next, we concatenate these features by channel. The formula is as follows:

$$f_{concat} = Concat(f_1', f_2', f_3', f_4', dim = channels). \tag{10}$$

The processed feature maps $f_1'$, $f_2'$, $f_3'$, and $f_4'$ are concatenated along the channel dimension to form a comprehensive feature map $f_{concat}$, restoring the original number of input channels. Finally, a 1×1 convolution kernel is applied to integrate the channel information, perform global fusion of fine-grained features, and generate the final output feature map $F_{\frac{1}{8}}$.The formula is as follows:

$$F_{\frac{1}{8}} = Conv_{1\times1}(f_{concat}). \tag{11}$$

The upsampled feature map is subsequently fused with the $I_{\frac{1}{4}}$ feature map. The process is formally expressed as follows:

$$F_{merge1} = Up(F_{\frac{1}{8}}) + I_{\frac{1}{4}} \tag{12}$$

where $Up(\cdot)$ denotes the upsampling operation, $I_{\frac{1}{4}}$ represents the feature map output by the encoder, and $F_{merge1}$ represents the weighted feature map. Next, position encoding is added to the feature map $F_{\frac{1}{8}}$ and the processed feature map is obtained through the Mamba2 module. The formula is as follows:

$$F_{merge1}^{'} = M2(F_{merge1} + P) \tag{13}$$

where $F_{merge1}^{'}$ represents the processed feature map, $M2$ represents the mamba2 model, and $P$ represents the positional encoding. Finally, the feature map $F_{merge1}^{'}$ is upsampled and then added to the feature map $I_{\frac{1}{2}}$ output by the encoder to obtain the final feature map with a scale of 1/2. The process can be expressed in mathematical form as follows:

$$F_{\frac{1}{2}} = Up(F_{merge1}') + I_{\frac{1}{2}}. \tag{14}$$

The processed feature map $F_{merge1}'$ is obtained through a second upsampling, followed by weighted fusion with the feature map $I_{\frac{1}{2}}$ output by the encoder to generate the final output feature map $F_{\frac{1}{2}}$.

## 3.2 Mamba2-enhanced Self- and Cross-Attention (MSC) Module

This module is primarily designed to strengthen the expressive power of image features. Its architectural overview is presented in Figure 4. Initially, positional encoding is incorporated into the feature map to embed spatial information explicitly, thereby improving the network's perception of positional relationships. Subsequently, the feature map is processed by a hybrid enhancement unit that

integrates a state-space model with a Transformer-based self-attention mechanism. This configuration combines the computational efficiency of state-space models with the global modeling strength of self-attention, leading to more robust and discriminative feature representations.

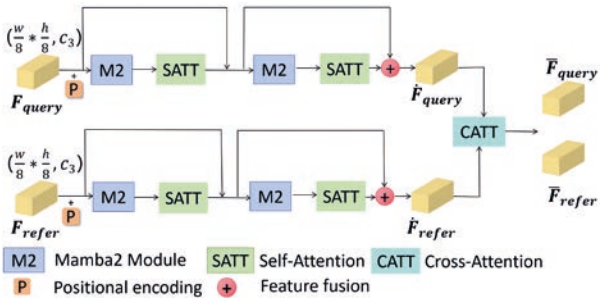


**Figure 4**. Detailed Structures of MSC Module within the MambaSC Framework. Positional encoding is first added to the feature map to embed spatial information. The features are then enhanced by a hybrid unit combining a state-space model and self-attention, followed by bidirectional information exchange via cross-attention between the two image features.

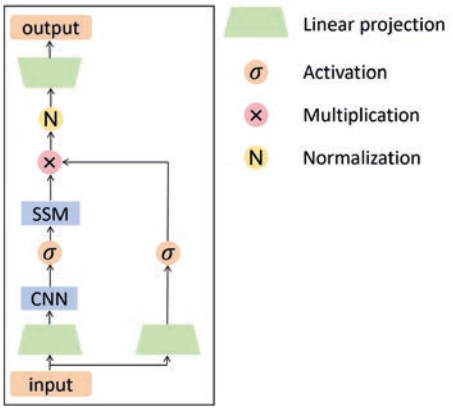


**Figure 5**. Detailed Structures of M2 within the MambaSC Framework. The input is linearly projected and processed by a CNN to extract local features. Then, an activation and an SSM capture long-range information. The CNN and SSM outputs are multiplied, normalized, and linearly projected to produce the final output.

Subsequently, the two image feature maps exchange bidirectional information through the cross-attention mechanism. This enables each image to extract complementary features from the other, enhancing its self-understanding and capturing richer cross-image correlations. This interaction fosters more comprehensive feature alignment and information sharing, providing a solid foundation for subsequent matching or fusion operations in multimodal image tasks.

The process is as follows: For the two input feature maps $F_{query}$ and $F_{refer}$ , positional encoding is first added to embed spatial position information explicitly. Each feature map is then processed through a module combining the state-space model and Transformer self-attention mechanism to generate enhanced feature maps $F'_{query}$ and $F'_{query}$. The original input information is retained via residual connections, resulting in $\dot{F}_{query}$ and $\dot{F}_{refer}$. In this process, the state-space model provides efficient sequence modeling, while the self-attention mechanism captures global context. The residual connection ensures stability during training and prevents feature degradation. The final outcome is feature maps $\dot{F}_{query}$ and $\dot{F}_{refer}$, which are more discriminative and globally aware.The formula for this part is as follows:

$$F'_{query} = SATT(Mamba2Block(F_{query} + P)) \quad (15)$$

$$F'_{refer} = SATT(Mamba2Block(F_{refer} + P)) \quad (16)$$

$$\dot{F}_{query} = F_{query} + F'_{query} \quad (17)$$

$$\dot{F}_{refer} = F_{refer} + F'_{refer}. \quad (18)$$

In this process, $P$ represents the positional encoding, which explicitly incorporates spatial information into the feature representations. The input features, $F_{query}$ and $F_{refer}$, are derived from the M2Backbone and serve as the initial embeddings for further processing. The self-attention mechanism, denoted as $SATT$, is applied to enhance these features by capturing long-range dependencies and contextual relationships within each image.

As a result of this operation, the transformed features, $F'_{query}$ and $F'_{refer}$ , are obtained. To preserve the original feature information while incorporating the enhanced representations, residual connections are introduced, leading to the updated feature maps $\dot{F}_{query}$ and $\dot{F}_{refer}$. These residual connections help maintain gradient flow during training, preventing vanishing gradient issues and ensuring stable optimization.

Following this step, the self-attention mechanism is applied once more to refine the representations further. This iterative refinement allows

the model to iteratively improve feature discriminability by reinforcing significant structures and suppressing irrelevant noise. The combined use of the Mamba module and Transformer-based self-attention enhances the model's capability to capture both local and global dependencies effectively. The mathematical formulations for this process are outlined below:

$$F''_{query} = SATT(Mamba2Block(\dot{F}_{query})) \quad (19)$$

$$F''_{refer} = SATT(Mamba2Block(\dot{F}_{refer})) \quad (20)$$

$$\ddot{F}_{query} = \dot{F}_{query} + F''_{query} \quad (21)$$

$$\ddot{F}_{refer} = \dot{F}_{refer} + F''_{refer}. \quad (22)$$

In this stage, $F''_{query}$ and $F''_{refer}$ represent the features that have undergone further refinement through the self-attention mechanism. These features capture long-range dependencies within their respective images, improving their contextual awareness and discriminability. The residual connections play a crucial role in preserving original feature information while integrating newly learned representations, resulting in the final enhanced features $\ddot{F}_{query}$ and $\ddot{F}_{refer}$.

To establish correspondences between the query and reference images, it is essential to enhance the interaction between $\ddot{F}_{query}$ and $\ddot{F}_{refer}$. Relying solely on independent feature refinement limits the ability of the model to understand cross-image relationships. Therefore, to address this, we introduce a cross-attention mechanism that facilitates direct information exchange between the two feature maps. This mechanism enables each feature map to attend to relevant regions in the corresponding image, effectively aligning structures and improving matching precision.

The cross-attention mechanism operates by treating one feature map as the query and the other as the key-value pair, allowing for bidirectional interaction between $\ddot{F}_{query}$ and $\ddot{F}_{refer}$. Specifically, the query feature map ($\ddot{F}_{query}$) is used to query information from the reference feature map ($\ddot{F}_{refer}$), and vice versa. This facilitates the exchange of information between corresponding regions, strengthening feature alignment. The mathematical formulation for this operation is given by:

$$\overline{F}_{query} = CATT(Q = \ddot{F}_{query}, K = \ddot{F}_{refer}, V = \ddot{F}_{refer}) \quad (23)$$

$$\overline{F}_{refer} = CATT(Q = \ddot{F}_{refer}, K = \ddot{F}_{query}, V = \ddot{F}_{query}) \quad (24)$$

where $CATT$ represents the cross-attention mechanism. In this process, the query ($Q$) comes from one feature map, while the key ($K$) and value ($V$) are derived from the other. This setup enables mutual refinement, allowing each feature map to adapt based on information from its counterpart.

The resulting feature maps, $\overline{F}_{query}$ and $\overline{F}_{refer}$, are the updated representations following cross-attention interaction. These refined features contain enhanced structural and semantic alignments between the query and reference images, making them well-prepared for subsequent matching operations. By effectively integrating complementary information, the cross-attention mechanism plays a vital role in improving matching robustness and accuracy.

These enhanced feature maps will then be utilized in the coarse-to-fine matching module, where they undergo further processing to establish reliable correspondences at multiple levels of granularity.

### 3.3 Coarse-to-Fine Matching Module

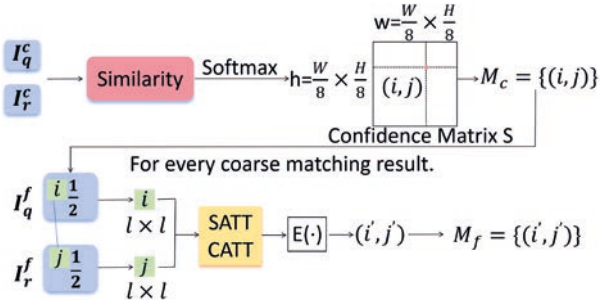


**Figure 6**. Coarse-to-Fine Matching Module. It first computes feature similarity between $I_q^c$ and $I_r^c$, applies Softmax to generate a confidence matrix S, and obtains coarse matches $M_c$. For each coarse match (i,j), local windows are cropped from fine features $I_q^f$ and $I_r^f$. These local features are enhanced via self-attention (SATT) and cross-attention (CATT) modules. The final refined match ($i'$, $j'$)is computed by function $E(\cdot)$, producing the fine match set $M_f$.

The Coarse-to-Fine Matching Module performs reliable feature matching through a hierarchical two-step strategy, beginning with coarse-level alignment and subsequently refining correspondences at a finer scale. The detailed workflow is illustrated in Figure 6.

### 3.3.1 Coarse Matching

To assess token-level similarity, we first construct a set of candidate correspondences by analyzing the interaction patterns between $I_q^c$ and $I_r^c$. The degree of match between tokens is then quantified through a bidirectional softmax operation, which computes the matching probabilities $P(i \leftrightarrow j)$ in both directions—from $I_q^c$ to $I_r^c$ and vice versa. This normalization step guarantees that the probability distributions along each axis are properly scaled (i.e., sum to 1), providing a reliable statistical foundation for incorporating dynamic reward and penalty strategies during training. The complete formulation is presented below:

$$P(i \leftrightarrow j) = \frac{softmax(S(i,\cdot))j \cdot softmax(S(\cdot,j))i}{\sum_{i^m,j^n} softmax(S(i^m,\cdot))j^n \cdot softmax(S(\cdot,j^n))i^m} \tag{25}$$

where the confidence matrix S is indexed by m and n, denoting its row and column positions, respectively. The matrix S encodes the matching confidence between feature tokens and serves as the basis for identifying reliable correspondences. By leveraging this matrix, unreliable matches can be effectively suppressed, resulting in a set of coarse-level correspondences denoted as $M_c$.

### 3.3.2 Fine Matching

Coarse matching yields an initial set of approximate matches, which are subsequently refined through fine matching for improved precision and alignment accuracy. In this stage, the query and reference feature maps at 1/2 resolution are denoted as $I_q^f$ and $I_r^f$ respectively, extracted from the feature map pair $F_{\frac{1}{2}}$. Initially, the coordinates of candidate matches identified during coarse matching are upsampled and projected onto $I_q^f$ and $I_r^f$ Around these projected locations, local windows are sampled to encapsulate both global context and fine-grained details. The extracted windows are then integrated to generate fused features $I_q^{fc}$ and $I_r^{fc}$, which are subsequently fed into an interaction module composed of dual attention mechanisms. This module enhances the expressiveness of feature representations and strengthens inter-feature dependencies. Finally, the refined correspondences $M_f$ are obtained by computing the expected values over the resulting matching distribution.

## 3.4 Supervision

### 3.4.1 Coarse Matching Loss

In the coarse matching phase, we adopt a supervised learning approach inspired by the policy gradient strategy, integrating a feedback-driven mechanism that rewards correct matches and penalizes errors to guide effective model learning. A threshold parameter u is introduced to guide the learning process, where adaptive reward and penalty signals are employed to refine the matching outcomes. A positive sample is rewarded with $\alpha$ once its maximum matching probability goes beyond the set threshold u. Conversely, a penalty is triggered either when a positive pair fails to meet the threshold or when a negative pair exceeds it. To maintain training stability and effectiveness, the penalty weight $\beta$ is dynamically modulated throughout training. The specific update rule for $\beta$ is defined as follows:

$$\beta = \begin{cases} 0, & \text{if epoch} < \text{n} \\ -0.01 \cdot (\text{epoch} - \text{n}), & \text{if n} \le \text{epoch} < \text{n} + 25 \\ -0.25, & \text{if epoch} \ge \text{n} + 25. \end{cases} \tag{26}$$

The coarse matching loss $L_1$ is formulated based on a policy gradient framework, aiming to improve the alignment between the query and reference features $I_q^c$ and $I_r^c$. This objective leverages the matching probability distribution $P(i \leftrightarrow j | I_q^c, I_r^c)$ alongside the gradient of the log-likelihood, $\nabla_\theta \phi_{ij}$, for each candidate correspondence $(i \leftrightarrow j)$. The loss function effectively captures the match quality, encouraging the model to learn discriminative and well-aligned feature representations. Let G denote the reward function, and the resulting policy gradient formulation is given as follows:

$$\nabla_\theta E_{M_{q\leftrightarrow r}} G_{(M_{q\leftrightarrow r})} = E_{I_q^c, I_r^c} \sum_{i,j} [P(i \leftrightarrow j | I_q^c, I_r^c, \theta_M) \cdot g(i \leftrightarrow j) \cdot \nabla_\theta \phi_{ij}] \tag{27}$$

$$\phi_{ij} = logP(i \leftrightarrow j | I_q^c, I_r^c, \theta_M) + logP(I_{q,i}^c | q, \theta_F) + logP(I_{r,j}^c | r, \theta_F) \tag{28}$$

where $\nabla_\theta E_{M_{q\leftrightarrow r}} G_{(M_{q\leftrightarrow r})}$ represents the gradient of the reward function $G_{(M_{q\leftrightarrow r})}$ with respect to the model parameters $\theta$ for the matching set, $E_{I_q^c, I_r^c}$ represents the expectation over all possible matching points, and $P(i \leftrightarrow j | I_q^c, I_r^c, \theta_M)$ denotes the probability distribution of the matches. $\theta_M$ refers to the

series of parameters in the matching model, while $g(i \leftrightarrow j)$ is the reward function that gives the reward value for matching feature points i and j. $\nabla_\theta \phi_{ij}$ represents the gradient of the logarithmic probability term $\phi_{ij}$ for the matching pair (i,j), and $\phi_{ij}$ denotes the total probability of feature matching. $logP(i \leftrightarrow j|I_q^c, I_r^c, \theta_M)$ represents the logarithmic probability of matching feature points i and j given the feature sets $I_q^c$ and $I_r^c$ and the matching model parameters $\theta_M$. $\theta_F$ refers to the series of parameters in the feature extraction model, while $logP(I_{q,i}^c|q, \theta_F)$ represents the logarithmic probability of feature point i given image q and feature extraction parameters $\theta_F$, reflecting the model's ability to correctly extract feature point i. Similarly, $logP(I_{r,j}^c|r, \theta_F)$ represents the logarithmic probability of feature point j given image r and feature extraction parameters $\theta_F$, reflecting the model's ability to correctly extract feature point j. The resulting objective function for coarse-level supervision is summarized below, in which t serves as the decision threshold for determining positive and negative matches.

$$L_1 = \nabla_\theta E_{M_{q\leftrightarrow r}} G_{(M_{q\leftrightarrow r})}, p(i \leftrightarrow j) > t. \tag{29}$$

### 3.4.2 Fine Matching Loss

To supervise the fine-grained regression process, the fine matching loss $L_2$ is introduced. It evaluates the accuracy of the predicted coordinates $(i + \nabla x, j + \nabla y)$ by comparing them with the ground-truth positions $(i_a, j_a)$. The offset error is averaged over all m matching points, facilitating precise alignment refinement. The $L_2$ loss is defined as:

$$L_2 = \frac{1}{m} \sum_{i=1}^{m} (i + \nabla x, j + \nabla y) - (i_a, j_a)^2. \tag{30}$$

### 3.4.3 Total Loss Function

The overall training objective integrates the losses from both coarse and fine matching stages, with each component weighted by the respective coefficients $\delta_1$ and $\delta_2$:

$$L_{total} = \delta_1 L_1 + \delta_2 L_2. \tag{31}$$

This unified loss function enables joint optimization, ensuring that both hierarchical stages contribute effectively to the model's learning and final matching accuracy.

# 4 Experiment

## 4.1 Dataset

To thoroughly evaluate the robustness and generalization ability of our proposed model, we conduct experiments on five multimodal datasets, each selected for its unique combination of scene types and image modalities. These datasets span a wide range of environments—from outdoor landscapes to indoor settings—and incorporate various modality combinations, including RGB, SAR, depth, near-infrared (NIR), short-wave infrared (SWIR), and optical imagery.

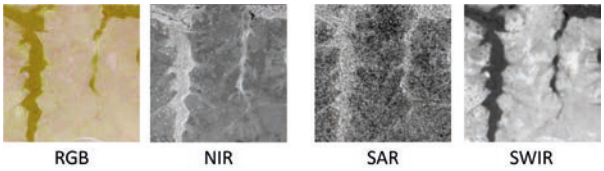


**Figure 7**. Example from the SEN12MS dataset: images of the same scene captured across different modalities.

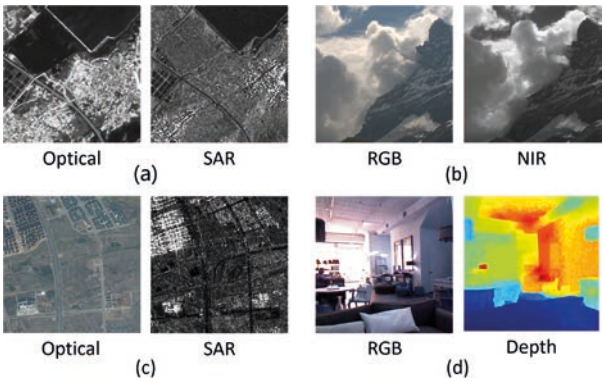


**Figure 8**. Examples from the remote sensing images of different scenes captured across various modalities. (a) Optical-SAR dataset. (b) RGB-NIR Scene dataset. (c) WHU-OPT-SAR dataset. (d) NYU-Depth V2 dataset.

1. **SEN12MS** [31]: SEN12MS offers rich multimodal satellite data, including RGB, SAR, NIR, and SWIR images. For model training, we utilize 16,000 image pairs, ensuring diverse cross-modality representation. To evaluate generalization capability, an additional 600 pairs are reserved exclusively for testing. The sample data diagram is shown in Figure 7.

2. **Optical-SAR**[32]: This dataset captures com-

plex geographic scenes such as harbors, plains, urban areas, and rivers, with each image pair consisting of optical and SAR modalities. All samples are standardized to a resolution of 512×512 pixels. We use 16,000 pairs for training and 500 pairs for testing, enabling evaluation under varied and real-world-like conditions. The sample data diagram is shown in Figure 8(a).

3. **RGB-NIR scene**[33]: This dataset explores the complementary information between RGB and NIR modalities in high-level scene understanding. It includes nine common scenarios such as streets, fields, and forests. We select 400 image pairs for training and retain 48 pairs for testing, focusing on evaluating the model's ability to generalize to unseen multimodal compositions. The sample data diagram is shown in Figure 8(b).

4. **WHU-OPT-SAR**[34]: Collected from Hubei Province, China, this dataset emphasizes six land cover categories: forest, road, water, village, farmland, and city. Due to the large size of raw images, we preprocess and crop them to construct 4,400 aligned pairs. Among these, 3,800 pairs are allocated for training, while 300 are used for validation and testing to ensure fair assessment across different data partitions. The sample data diagram is shown in Figure 8(c).

5. **NYU-Depth V2**[35]: Targeting indoor scene understanding, this dataset comprises 1,449 RGB-depth image pairs collected from densely annotated video sequences. We assign 1,049 pairs for training to help the model learn structural and spatial patterns specific to indoor environments, while the remaining 400 pairs are used to benchmark testing performance. The sample data diagram is shown in Figure 8(d).

## 4.2 Implementation Details

**Table 1**. MambaSC hyperparameter settings.

| Hyperparameters | Settings |
|---|---|
| M2 configuration in M2backbone | Layer=4,head=64 |
| M2-Elaborate configuration in M2backbone | Layer=4,head=16 |
| M2 configuration in MSC | Layer=8,head=64 |
| MSC module self-attention(SATT) layer number | 4 |
| MSC module cross-attention(CATT) layer number | 8 |
| Coarse matching threshold | 0.2 |
| Fine matching module SATT&CATT attention layer | 1 |
| Training image size | 320 |

During training, we adopted the AdamW optimizer [36] combined with the CosineAnnealingLR scheduler [37] to promote stable convergence and enhance model performance. The initial learning rate was set to $1 * 10^{-4}$, and was gradually decreased following a cosine decay schedule.

In the M2 backbone architecture, feature maps at four hierarchical scales (1/2, 1/4, 1/8, and 1/16) were configured with channel dimensions of 128, 192, 256, and 512, respectively. To maintain a balance between computational efficiency and memory consumption, the batch size was fixed at 4, and a confidence threshold t = 0.2 was applied during the matching phase.

For training supervision, we assigned equal weights to both coarse and fine matching losses, with $\delta_1 = \delta_2 = 1$, unless otherwise specified. These coefficients can be flexibly tuned for balancing training dynamics or incorporating task-specific emphasis.

All experiments were conducted on a single NVIDIA RTX 3090 GPU. To accommodate the varying characteristics of different datasets, we slightly adjusted some module-specific hyperparameters in MambaSC. Detailed configurations for each dataset are summarized in Table 1.

## 4.3 Evaluation Indicators

We systematically compared our proposed method with several cutting-edge approaches to demonstrate its effectiveness in multimodal image feature matching. The benchmarked algorithms include UFM [38], AspanFormer [9], FmCFA [39], TopicFM [8], QuadTree [18], Matchformer [10], MIVI [40], FeMIP [32], LoFTR [7], TFeat [41], MatchosNet [42], and MatchNet [43]. For a fair and rigorous evaluation, all models were trained on the same datasets using identical experimental settings, and their performance was assessed using the same test benchmarks.

Homography Estimation: For the homography estimation evaluation, we follow the experimental protocol described in [7]. The initial homography matrix is obtained using OpenCV and subsequently refined with RANSAC to enhance robustness. Each test case involves multimodal remote sensing image pairs captured by different sensors. Consistent with the procedure in [4], the angular deviation between

the estimated homography $\hat{H}$ and the ground truth $H$ is measured by averaging the reprojection errors of the image corner points. The evaluation metric is defined as the area under the curve (AUC) computed at pixel thresholds of 5, 10, and 20.

Mean Matching Accuracy (MMA): MMA [44] evaluates feature matching reliability by preserving mutual nearest-neighbor correspondences within each image pair. A correspondence is regarded as correct if its reprojection error, derived from the estimated homography, is below a given pixel threshold. The performance curve is then drawn by plotting matching accuracy against different thresholds, showing how the accuracy changes with increasing tolerance. The MMA score represents the mean proportion of correct matches across all test pairs at each threshold, providing a holistic measure of overall matching performance.

Mean Distance Error (MDE) Test: To further examine the localization accuracy of various matching methods, we compute the mean spatial errors of corresponding keypoints. Only pairs whose horizontal and vertical deviations are both less than 1 pixel are retained as valid matches. In this assessment, horizontal error, vertical error, and total Euclidean distance error are denoted as $H_{\text{rmse}}$, $V_{\text{rmse}}$, and $HV_{\text{rmse}}$, respectively. We report the average displacements along both axes and the overall mean distance error of all valid correspondences for each evaluated approach. The formal definitions of $H_{\text{rmse}}$, $V_{\text{rmse}}$, and $HV_{\text{rmse}}$ are provided in the following:

$$H_{\text{rmse}} = \sqrt{\frac{1}{N}\sum_{i}(a_i^{x1} - a_i^{x2})^2} \tag{32}$$

$$V_{\text{rmse}} = \sqrt{\frac{1}{N}\sum_{i}(b_i^{x1} - b_i^{x2})^2} \tag{33}$$

$$HV_{\text{rmse}} = \sqrt{\frac{1}{N}\sum_{i}\left((a_i^{x1} - a_i^{x2})^2 + (b_i^{x1} - b_i^{x2})^2\right)} \tag{34}$$

where $(a_i^{x1}, b_i^{x1})$ denotes the coordinates of a point in the $x1$-modality image, and $(a_i^{x2}, b_i^{x2})$ represents its corresponding point in the $x2$-modality image. $N$ refers to the total number of matched point pairs.

## 4.4 Homography Estimation Experiment

Within the homography estimation framework, we benchmarked MambaSC against several representative methods, including UFM [38], FmCFA [39], MIVI [40], FeMIP [32], LoFTR[7], AspanFormer [9], TopicFM [8], and MatchosNet [42]. Homography matrices were estimated using Pydegensac in conjunction with RANSAC to ensure robustness. Following the evaluation protocol described in [5], we report the AUC metric based on cumulative angular corner errors, computed at thresholds of 5, 10, and 20 pixels to accommodate the inherent complexity of multimodal remote sensing imagery.

To comprehensively assess cross-modal performance, four representative image pair types were selected from the SEN12MS dataset for detailed analysis: NIR-RGB, NIR-SWIR, SAR-NIR and SAR-SWIR. As shown in Table 2, MambaSC consistently outperforms other methods across all modality combinations, underscoring its strong ability to handle heterogeneous input pairs and achieve precise cross-modal alignment.

Furthermore, to examine the generalization capability of the proposed framework, we extended the evaluation to several diverse datasets, including Optical-SAR, RGB-NIR, WHU-OPT-SAR, and NYU-Depth V2. As presented in Table 3, MambaSC achieves the leading performance in most cases, demonstrating its robustness and adaptability to varying modalities, scenes and data distributions. Collectively, these results affirm the reliability and effectiveness of MambaSC in real-world homography estimation tasks.

## 4.5 Evaluation of the Number of Correct Matches

We define NCM (Number of Correct Matches) as the count of matches with a Root Mean Square Error (RMSE) below three pixels, SR (Success Rate) as the ratio of successful match estimations to total trials, and RCM (Ratio of Correct Matches) as the proportion of correct correspondences among all matches.

As shown in Table 4, average NCM values vary considerably across datasets due to modality differences. MambaSC achieves the highest average NCM on all datasets, demonstrating consistent su-

**Table 2**. Homography estimation results on the SEN12MS dataset.

| Different Datasets | Method | Homography est. AUC | | |
|---|---|---|---|---|
| | | @5px | @10px | @20px |
| NIR-RGB | MatchosNet [42] | 23.17 | 49.08 | 60.54 |
| | LoFTR [7] | 21.54 | 43.07 | 55.36 |
| | FmCFA [39] | 27.62 | 47.98 | 66.58 |
| | FeMIP [32] | 25.92 | 46.41 | 62.25 |
| | MIVI [40] | 29.58 | 46.35 | 58.77 |
| | AspanFormer [9] | 46.51 | 63.76 | 78.35 |
| | TopicFM [8] | 53.25 | 66.59 | 81.43 |
| | UFM [38] | 57.09 | 72.88 | 82.28 |
| | **MambaSC (Ours)** | **58.66** | **73.28** | **83.65** |
| SAR-SWIR | MatchosNet [42] | 18.05 | 29.74 | 30.36 |
| | LoFTR [7] | 28.44 | 42.39 | 45.38 |
| | FmCFA [39] | 9.60 | 20.75 | 37.39 |
| | FeMIP [32] | 29.31 | 43.30 | 48.65 |
| | AspanFormer [9] | 31.47 | 44.81 | 49.22 |
| | TopicFM [8] | 32.88 | 45.60 | 51.30 |
| | UFM [38] | 17.74 | 26.24 | 45.31 |
| | **MambaSC (Ours)** | **33.67** | **47.18** | **53.96** |
| SAR-NIR | MatchosNet [42] | 19.84 | 30.69 | 42.88 |
| | LoFTR [7] | 31.58 | 39.98 | 50.36 |
| | FmCFA [39] | 33.15 | 41.23 | 53.49 |
| | FeMIP [32] | 33.27 | 43.88 | 52.10 |
| | AspanFormer [9] | 33.59 | 45.60 | 54.71 |
| | TopicFM [8] | 34.09 | 47.22 | 57.20 |
| | UFM [38] | 25.24 | 35.31 | 50.48 |
| | **MambaSC (Ours)** | **35.77** | **48.26** | **59.25** |
| NIR-SWIR | MatchosNet [42] | 20.07 | 32.21 | 43.05 |
| | LoFTR [7] | 34.58 | 44.83 | 53.60 |
| | MIVI [40] | 35.51 | 45.41 | 53.75 |
| | FmCFA [39] | 35.27 | 45.11 | 53.02 |
| | AspanFormer [9] | 36.87 | 47.16 | 55.74 |
| | TopicFM [8] | 36.55 | 49.32 | 58.06 |
| | UFM [38] | 35.76 | 46.52 | 54.91 |
| | **MambaSC (Ours)** | **37.21** | **50.22** | **60.06** |

**Table 3**. Homography Estimation on Multimodal Datasets.

| Different Datasets | Method | Homography est. AUC | | |
|---|---|---|---|---|
| | | @5px | @10px | @20px |
| Optical-SAR | LoFTR [7] | 16.59 | 39.29 | 56.97 |
| | MatchosNet [42] | 32.33 | 53.40 | 66.37 |
| | FeMIP [32] | 54.07 | 69.38 | 77.62 |
| | MIVI [40] | 55.13 | 64.54 | 79.22 |
| | FmCFA [39] | 55.10 | 69.99 | 80.11 |
| | AspanFormer [9] | 54.21 | 69.97 | 79.22 |
| | TopicFM [8] | 55.02 | 70.30 | 81.39 |
| | UFM [38] | **56.20** | **73.14** | **83.42** |
| | **MambaSC (Ours)** | 55.61 | 71.20 | 82.29 |
| RGB-NIR Scene | LoFTR [7] | 40.64 | 60.21 | 75.39 |
| | MatchosNet [42] | 35.02 | 50.33 | 64.83 |
| | FeMIP [32] | 43.76 | 62.50 | 78.49 |
| | MIVI [40] | 16.74 | 39.71 | 61.80 |
| | FmCFA [39] | 45.84 | 63.21 | 79.59 |
| | AspanFormer [9] | 43.69 | 62.88 | 80.91 |
| | TopicFM [8] | 45.11 | 63.57 | 81.66 |
| | UFM [38] | 45.93 | 64.66 | 80.07 |
| | **MambaSC (Ours)** | **48.88** | **70.65** | **83.81** |
| WHU-OPT-SAR | LoFTR [7] | 16.18 | 33.33 | 53.61 |
| | MatchosNet [42] | 24.79 | 52.44 | 72.58 |
| | FeMIP [32] | 27.82 | 50.77 | 71.56 |
| | MIVI [40] | 26.07 | 48.40 | 70.74 |
| | FmCFA [39] | 35.26 | 44.39 | 58.03 |
| | AspanFormer [9] | 39.59 | 52.63 | 64.70 |
| | TopicFM [8] | 40.32 | 54.80 | 68.36 |
| | UFM [38] | **40.83** | 58.11 | 71.15 |
| | **MambaSC (Ours)** | 40.52 | **58.63** | **74.59** |
| NYU-Depth V2 | LoFTR [7] | 28.87 | 44.07 | 58.73 |
| | MatchosNet [42] | 21.58 | 45.79 | 67.90 |
| | FeMIP [32] | 43.06 | 58.14 | 72.35 |
| | MIVI [40] | 44.07 | 58.15 | 73.36 |
| | FmCFA [39] | 23.61 | 45.32 | 66.73 |
| | AspanFormer [9] | 52.74 | 60.41 | 71.51 |
| | TopicFM [8] | 55.82 | 64.73 | 72.67 |
| | UFM [38] | 58.67 | 65.22 | 75.96 |
| | **MambaSC (Ours)** | **61.24** | **68.51** | **79.45** |

periority. Datasets involving SAR imagery exhibit notably lower NCM scores, which aligns with prior observations that SAR images contain fewer textures, increasing matching difficulty.

Table 5 presents the RCM evaluation. All detector-free methods—from LoFTR to MambaSC—employ fine-level filtering to remove mismatches, yielding low false match rates. Among them, MambaSC consistently achieves the highest RCM across all test cases. On the Optical-SAR dataset, large scale differences between modalities amplify reprojection errors when RMSE is computed on the higher-resolution optical image, leading to an increased mismatch rate.

Table 6 summarizes SR performance. Detector-free methods show higher SR due to their precise, descriptor-free matching mechanisms, while detector-based methods such as MatchosNet display slightly lower SR. MambaSC attains a perfect SR (100%) across all datasets, highlighting its exceptional robustness in diverse multimodal matching scenarios.

## 4.6 Matching Accuracy Experiment

In the evaluation of average matching accuracy, we conducted a comprehensive comparison between MambaSC and a set of well-established baselines, including FmCFA [39], FeMIP [32], AspanFormer [9], TopicFM [8], LoFTR[7], TFeat [41], and MatchosNet [42]. For targeted analysis, we focused on four particularly challenging modality combinations within the SEN12MS dataset: NIR-RGB, SAR-SWIR, SAR-NIR, and NIR-SWIR. The results, summarized in Figure 9, highlight the robustness of MambaSC in handling high-modality discrepancies.

To further assess its applicability across broader domains, we extended the experiment to additional multimodal datasets, including the RGB-NIR Scene dataset, NYU-Depth V2, Optical-SAR, and WHU-OPT-SAR. As shown in Figure 10, the evaluation plots illustrate pixel-level thresholds on the x-axis and the corresponding correct match ratios on the y-axis, facilitating a detailed performance comparison of the models across a range of spatial tolerance levels. MambaSC consistently maintains a strong advantage across diverse scenarios and modalities.

The results clearly demonstrate that methods such as MambaSC, FmCFA [39], FeMIP [32], and TopicFM [8] exhibit strong performance, outperforming many competing approaches. Among these, MambaSC consistently achieves high matching accuracy across various scenarios, highlighting its robustness and effectiveness.

## 4.7 Multimodal Image Matching Visualization

To assess the effectiveness of MambaSC, we conducted comparisons with state-of-the-art methods, including LoFTR[7], FeMIP [32], and FmCFA [39], across multiple cross-modal datasets such as SEN12MS, NYU Depth V2, Optical-SAR, RGB-NIR Scene, and WHU-OPT-SAR. Matching points were evaluated using the OpenCV RANSAC algorithm with a reprojection threshold of 3 pixels; matches below this threshold were regarded as correct and visualized with green lines. As shown in Figure 11, MambaSC achieves superior matching accuracy and robustness across all evaluated modalities, significantly surpassing competing approaches.

## 4.8 Evaluation of computational cost

In this study, we evaluate the model parameters and computational efficiency of MambaSC on the NYU-Depth V2 dataset using an RTX 3090 GPU. As shown in Table 7, MambaSC exhibits a lightweight architecture, with a parameter count comparable to FeMIP and UFM, yet significantly lower than FmCFA. In terms of efficiency, MambaSC requires only 1052 MB of memory during inference—much less than competing models—and completes prediction in 42 seconds, nearly twice as fast as FmCFA (87 seconds). In homography estimation, MambaSC achieves 61.24% accuracy at the 5-pixel threshold, demonstrating its ability to maintain high matching performance while substantially reducing computational complexity. These results confirm that MambaSC effectively balances accuracy and efficiency, making it well-suited for resource-constrained applications.

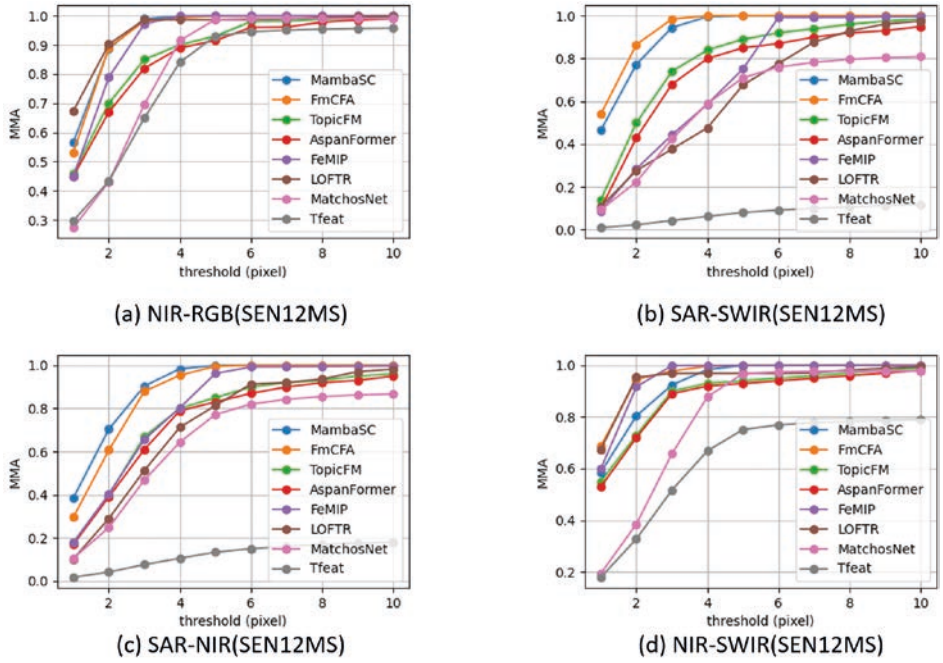


**Figure 9**. Cross-Modal Average Matching Accuracy Evaluation on SEN12MS Dataset.

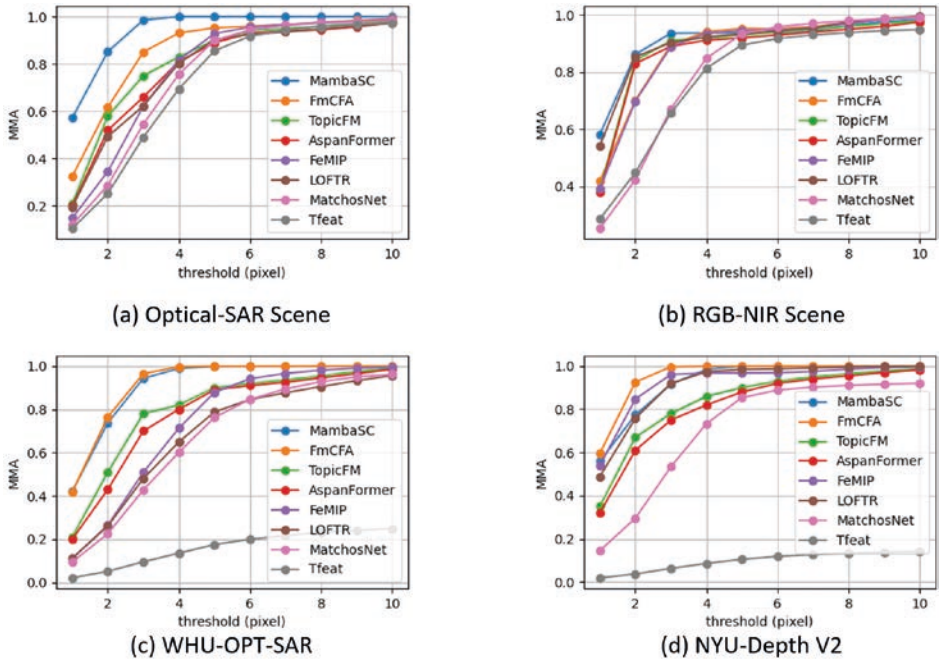


**Figure 10**. Cross-Modal Matching Accuracy Evaluation: (a) Optical-SAR Scenes, (b) RGB-NIR Scenes, (c) WHU-OPT-SAR, (d) NYU-Depth V2.

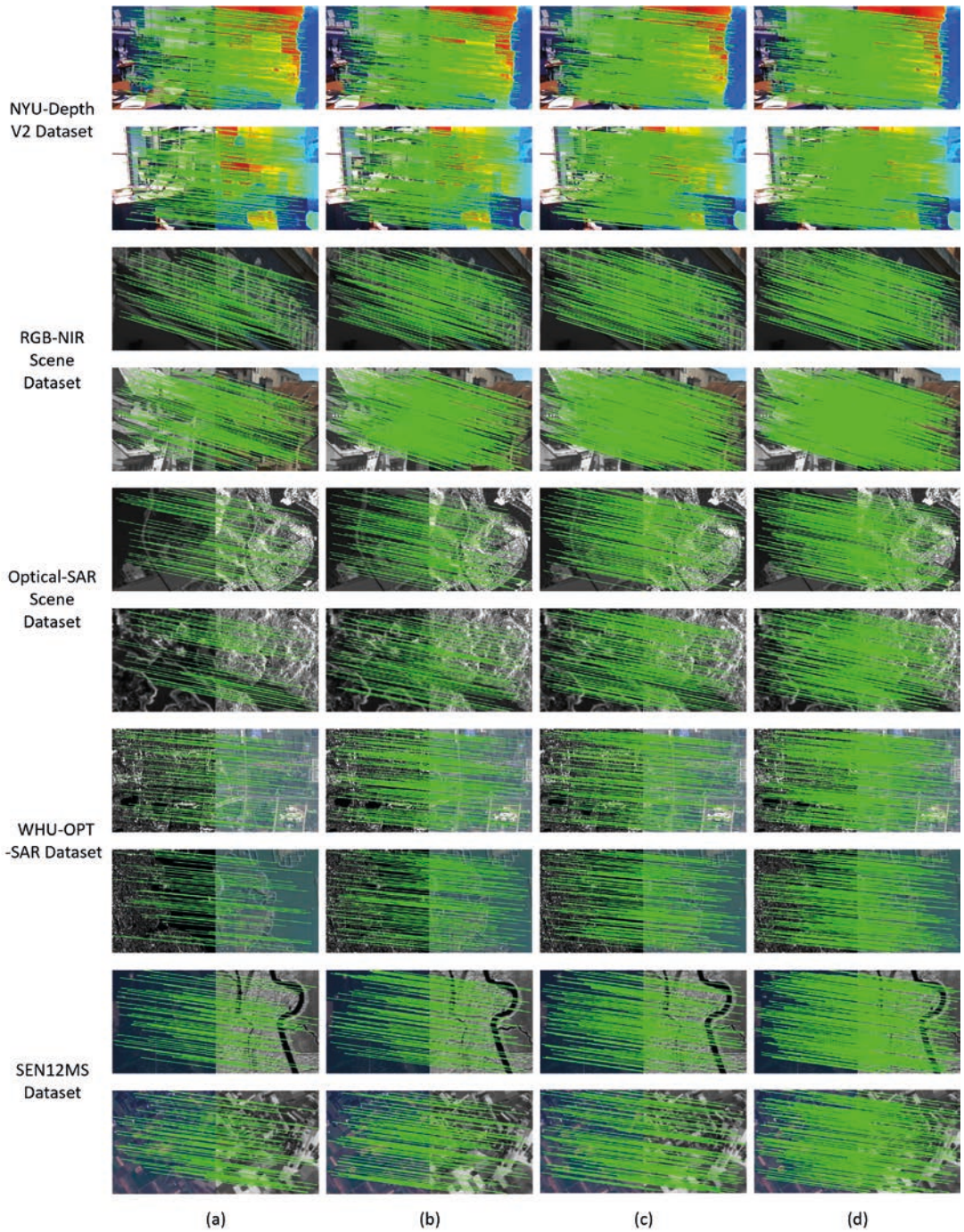


**Figure 11**. Visualization of multimodal remote sensing image feature matching. (a) LoFTR. (b) FeMIP. (c) FmCFA. (d) MambaSC.

**Table 4**. Evaluation of the Average of the NCM for Multimodal Remote Sensing Images of Different Scenes.

| Method | SEN12MS Dataset | | | | Optical-SAR | RGB-NIR Scene | WHU-OPT-SAR |
|---|---|---|---|---|---|---|---|
| | NIR-OPT | SAR-SWIR | SAR-NIR | NIR-SWIR | | | |
| MatchosNet [42] | 2530.21 | 178.92 | 120.81 | 1208.45 | 56.31 | 1302.82 | 187.49 |
| LoFTR [7] | 3175.78 | 287.95 | 211.76 | 1854.32 | 127.36 | 1753.13 | 261.27 |
| QuadTree [18] | 3322.97 | 298.76 | 237.65 | 1987.66 | 119.25 | 1846.01 | 591.45 |
| Matchformer [10] | 3495.83 | 310.00 | 307.25 | 2279.58 | 135.37 | 1938.21 | 1012.93 |
| FeMIP [32] | 3421.01 | 305.72 | 396.63 | 2201.98 | 169.31 | 1768.42 | 897.64 |
| AspanFomer [9] | 3601.22 | 409.73 | 480.61 | 2597.22 | 205.42 | 2085.43 | 897.64 |
| TopicFM [8] | 3587.46 | 418.73 | 498.20 | 2658.04 | 208.31 | 2198.42 | 1875.69 |
| **MambaSC** | **3609.24** | **420.21** | **500.28** | **2666.46** | **210.63** | **2201.89** | **1885.57** |

**Table 5**. Evaluation of RCM for Multimodal Remote Sensing Images of Different Scenes.

| Method | SEN12MS Dataset | | | | Optical-SAR | RGB-NIR Scene | WHU-OPT-SAR |
|---|---|---|---|---|---|---|---|
| | NIR-OPT | SAR-SWIR | SAR-NIR | NIR-SWIR | | | |
| MatchosNet [42] | 72.06% | 69.17% | 76.33% | 78.30% | 26.18% | 70.06% | 66.35% |
| LoFTR [7] | 83.09% | 78.26% | 80.40% | 81.25% | 31.99% | 76.26% | 74.91% |
| QuadTree [18] | 85.44% | 79.66% | 80.27% | 85.30% | 32.11% | 78.80% | 77.37% |
| Matchformer [10] | 89.92% | 84.74% | 85.01% | 87.40% | 36.13% | 82.09% | 80.96% |
| FeMIP [32] | 90.05% | 85.70% | 84.59% | 88.33% | 38.21% | 84.35% | 83.66% |
| AspanFomer [9] | 95.02% | 88.34% | 90.36% | 91.12% | 40.24% | 89.87% | 85.91% |
| TopicFM [8] | 97.55% | 91.49% | 92.88% | 93.89% | 42.70% | 90.66% | 86.75% |
| **MambaSC** | **97.63%** | **92.29%** | **94.57%** | **94.59%** | **45.21%** | **91.38%** | **87.56%** |

**Table 7**. Model Performance Comparison: Architecture Efficiency vs. Accuracy.

| Model (Year) | Params (M) | Memory (MB) | Time (s) | AUC@5px (%) |
|---|---|---|---|---|
| FeMIP [32] | **36.46** | 1550 | 47 | 43.06 |
| MIVI [40] | 47.29 | 1714 | 96 | 44.07 |
| FmCFA [39] | 41.06 | 1654 | 87 | 23.61 |
| UFM [38] | 36.86 | 1472 | **41** | 58.67 |
| MambaSC | 37.03 | **1052** | 42 | **61.24** |

**Table 8**. Evaluation of matching speed and resources on the RGB-NIR Scene dataset.

| Method | Frames Per Second (FPS) ↑ | Storage (MB) ↓ |
|---|---|---|
| Tfeat [41] | 1.17 | 0.334 |
| MatchosNet [42] | 1.30 | 0.368 |
| LoFTR [7] | 3.59 | 34.780 |
| FeMIP [32] | 4.80 | 34.782 |
| UFM [38] | 13.56 | 20.666 |
| MambaSC | **13.96** | **18.372** |

To further assess computational cost, we compared MambaSC with several state-of-the-art matching methods on the RGB-NIR Scene dataset, also using an RTX 3090 GPU. Matching speed was measured in frames per second (FPS), and resource usage in storage (MB). As shown in Table 8, MambaSC achieves the fastest matching speed among all compared methods. Among the semi-dense approaches (MambaSC, UFM, LoFTR, and FeMIP), which typically require higher computational resources than sparse methods, MambaSC demonstrates the best trade-off, achieving superior matching accuracy and speed with the lowest resource consumption.

## 4.9 Ablation Experiment

To assess the effectiveness of each component within our proposed framework, we conducted comprehensive ablation experiments on the NYU-Depth V2 dataset.

As summarized in Table 9, four experimental configurations were designed to progressively analyze the contribution of each module. The first configuration employed a standard Feature Pyramid Network (FPN) combined with a basic feature optimization module [7], serving as the baseline without incorporating the Mamba2 component. Subsequently, we introduced the M2 and M2-Elaborate components into the M2backbone, as well as the M2 and S&C-Att (Self-Attention and Cross-Attention) components into the MSC module, to evaluate their respective effects.

**Table 6**. Evaluation of SR for Multimodal Remote Sensing Images of Different Scenes.

| Method | SEN12MS Dataset | | | | Optical-SAR | RGB-NIR Scene | WHU-OPT-SAR |
|---|---|---|---|---|---|---|---|
| | NIR-OPT | SAR-SWIR | SAR-NIR | NIR-SWIR | | | |
| MatchosNet [42] | 85.24% | 74.35% | 69.91% | 78.25% | 69.39% | 88.95% | 75.86% |
| LoFTR [7] | 97.84% | 93.63% | 95.01% | 93.88% | 93.06% | 96.33% | 93.72% |
| QuadTree [18] | 96.06% | 92.88% | 94.60% | 94.47% | 92.75% | 91.24% | 92.38% |
| Matchformer [10] | 100% | 98.33% | 97.97% | 98.50% | 95.00% | 94.38% | 97.73% |
| FeMIP [32] | 99.82% | 97.95% | 100% | 98.30% | 94.21% | 95.27% | 96.96% |
| AspanFomer [9] | 100% | 100% | 98.85% | 100% | 95.43% | 100% | 100% |
| TopicFM [8] | 100% | 99.12% | 100% | 100% | 99.87% | 100% | 100% |
| **MambaSC** | **100%** | **100%** | **100%** | **100%** | **99.96%** | **100%** | **100%** |

The results indicate that both the M2 and M2-Elaborate components in the M2backbone significantly enhance feature representation and overall matching accuracy. Interestingly, in the MSC module, utilizing self-attention and cross-attention independently yields greater accuracy improvement than applying the Mamba2 component alone. Nevertheless, their combination produces a substantial performance boost. This phenomenon can be attributed to the complementary nature of the two mechanisms: while self-attention and cross-attention effectively capture global dependencies and inter-modal feature interactions, the Mamba2 component provides efficient sequence modeling and dynamic feature fusion capabilities. Their synergistic integration achieves an optimal balance between local and global context modeling, resulting in superior cross-modal semantic alignment and overall performance.

**Table 9**. Impact of Module Variants on Matching Accuracy at Different Pixel Thresholds.

| Dataset | M2-backbone | | MSC Module | | Homography est. AUC | | |
|---|---|---|---|---|---|---|---|
| | M2 | M2-Elaborate | M2 | S&C-ATT | @5px | @10px | @20px |
| NYU-Depth V2 | × | × | × | × | 36.81 | 45.92 | 58.17 |
| | ✓ | × | × | × | 42.36 | 53.74 | 64.08 |
| | × | ✓ | × | × | 44.90 | 55.63 | 66.42 |
| | × | × | ✓ | × | 45.62 | 56.75 | 67.29 |
| | × | × | × | ✓ | 56.18 | 64.27 | 74.86 |
| | ✓ | ✓ | × | × | 48.27 | 59.16 | 69.35 |
| | ✓ | × | ✓ | × | 49.85 | 60.92 | 70.89 |
| | ✓ | × | × | ✓ | 57.68 | 65.84 | 75.62 |
| | × | ✓ | ✓ | × | 52.41 | 62.30 | 72.11 |
| | × | ✓ | × | ✓ | 59.83 | 67.45 | 77.18 |
| | × | × | ✓ | ✓ | 58.72 | 66.54 | 76.85 |
| | ✓ | ✓ | ✓ | × | 55.92 | 64.38 | 74.12 |
| | ✓ | ✓ | × | ✓ | 61.10 | 68.51 | 78.62 |
| | ✓ | × | ✓ | ✓ | 60.24 | 67.75 | 77.85 |
| | × | ✓ | ✓ | ✓ | 60.78 | 68.24 | 78.12 |
| | ✓ | ✓ | ✓ | ✓ | **61.24** | **68.51** | **79.45** |

Furthermore, we conducted additional ablation experiments to analyze the influence of model scale on performance. As shown in the Table 10, increasing the number of layers and heads within the M2 and MSC modules consistently improves the results, suggesting that deeper architectures enhance the model's capacity for feature extraction and representation learning.

**Table 10**. Performance on NYU-Depth V2 dataset under different hyperparameter settings.

| Configuration | | | Homography est. AUC | | |
|---|---|---|---|---|---|
| Setting | Component | Parameter | @5px | @10px | @20px |
| #1 | M2 in M2backbone | Layer = 2, Head = 16 | 40.71 | 49.52 | 62.26 |
| | M2-Elab in M2backbone | Layer = 2, Head = 4 | | | |
| | M2 in MSC | Layer = 4, Head = 16 | | | |
| | SATT in MSC | Layer = 2, Head = 16 | | | |
| | CATT in MSC | Layer = 4, Head = 16 | | | |
| #2 | M2 in M2backbone | Layer = 3, Head = 32 | 55.68 | 64.45 | 73.18 |
| | M2-Elab in M2backbone | Layer = 3, Head = 8 | | | |
| | MSC-M2 | Layer = 6, Head = 32 | | | |
| | SATT in MSC | Layer = 3, Head = 32 | | | |
| | CATT in MSC | Layer = 6, Head = 32 | | | |
| #3 | M2-backbone | Layer = 4, Head = 64 | **61.24** | **68.51** | **79.45** |
| | M2-Elab | Layer = 4, Head = 16 | | | |
| | MSC-M2 | Layer = 8, Head = 64 | | | |
| | SATT in MSC | Layer = 4, Head = 64 | | | |
| | CATT in MSC | Layer = 8, Head = 64 | | | |

# 5 Conclusion

In this paper, we propose a novel multimodal image matching framework, MambaSC, which leverages a multi-scale architecture, M2Backbone, combined with a refined attention mechanism. This design enables more effective and robust feature representation across heterogeneous image modalities. Additionally, we designed a novel feature enhancement module, Mamba2 with Self and Cross Attention Module (MSC Module), which dynamically enhances image features by leveraging both self-attention and cross-attention mechanisms. Extensive experiments conducted on various multimodal datasets demonstrate that MambaSC delivers superior matching performance, validating its effectiveness and robustness in real-world applications.

## 6 Data Availability

The datasets utilized in this study are available through publicly accessible repositories listed below.
SEN12MS dataset:
https://med iatum.ub.tum.de/1474000
Optical-SAR dataset:
https://www.kaggle.com/datasets/requiemonk/sentinel12-image-pairs-segregated-by-terrain/data
WHU-OPT-SAR dataset:
https://github.com/AmberHen/WHU-OPT-SAR-dataset
RGB-NIR scene dataset:
https://www.epfl.ch/labs/ivrl/research/downloads/rgb-nir-scene-dataset/
NYU-Depth V2 dataset:
https://www.kaggle.com/datasets/soumikrakshit/nyu-depth-v2

## References

[1] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33:978–994, 2010.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In European Conference on Computer Vision, pages 404–417. Springer, Berlin, Heidelberg, 2006.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In 2011 International Conference on Computer Vision, pages 2564–2571. IEEE, 2011.

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 224–236, 2018.

[5] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4938–4947, 2020.

[6] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 17627–17638, 2023.

[7] J. Sun, Z. Shen, Y. Wang, et al. Loftr: Detector-free local feature matching with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8922–8931, 2021.

[8] Khang Truong Giang, Soohwan Song, and Sungho Jo. Topicfm: Robust and interpretable topic-assisted feature matching. In Proceedings of the AAAI conference on artificial intelligence, volume 37, pages 2447–2455, 2023.

[9] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In European conference on computer vision, pages 20–36. Springer, 2022.

[10] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In Proceedings of the Asian conference on computer vision, pages 2746–2762, 2022.

[11] J. Edstedt, I. Athanasiadis, M. Wadenbäck, et al. Dkm: Dense kernelized feature matching for geometry estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 17765–17775, 2023.

[12] J. Edstedt, Q. Sun, G. Bökman, et al. Roma: Revisiting robust losses for dense feature matching. arXiv preprint arXiv:2305.15404, 2023.

[13] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. IEEE transactions on neural networks, 20(1):61–80, 2008.

[14] Hanwen Jiang, Arjun Karpur, Bingyi Cao, Qixing Huang, and Andre Araujo. Omniglue: Generalizable feature matching with foundation model guidance. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 19865–19875, 2024.

[15] Johan Edstedt, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Dedode: Detect, don't describe—describe, don't detect for local feature matching. In 2024 International Conference on 3D Vision (3DV), pages 148–157. IEEE, 2024.

[16] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient loftr: Semi-dense local feature matching with sparse-like speed. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 21666–21675, 2024.

[17] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. In Advances in Neural Information Processing Systems, volume 33, pages 14254–14265, 2020.

[18] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. arXiv preprint arXiv:2201.02767, 2022.

[19] P. Truong, M. Danelljan, L. Van Gool, et al. Learning accurate dense correspondences and when to trust them. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5714–5724, 2021.

[20] Tao Xie, Kun Dai, Ke Wang, Ruifeng Li, and Lijun Zhao. Deepmatcher: a deep transformer-based network for robust and accurate local feature matching. Expert Systems with Applications, 237:121361, 2024.

[21] Yongxian Zhang, Chaozhen Lan, Haiming Zhang, Guorui Ma, and Heng Li. Multimodal remote sensing image matching via learning features and attention mechanism. IEEE Transactions on Geoscience and Remote Sensing, 62:1–20, 2024.

[22] Wang Zhang, Tingting Li, Yuntian Zhang, Gensheng Pei, Xiruo Jiang, and Yazhou Yao. Ltformer: A light-weight transformer-based self-supervised matching network for heterogeneous remote sensing images. Information Fusion, 109:102425, 2024.

[23] Xin Hu, Yan Wu, Zhikang Li, Zhifei Yang, and Ming Li. Multi-feature alignment and matching network for sar and optical image registration. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2024.

[24] Yongjun Zhang, Peihao Wu, Yongxiang Yao, Yi Wan, Wenfei Zhang, Yansheng Li, and Xiaohu Yan. Multi-modal remote sensing image robust matching based on second-order tensor orientation feature transformation. IEEE Transactions on Geoscience and Remote Sensing, 2025.

[25] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396, 2021.

[26] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752, 2023.

[27] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. arXiv preprint arXiv:2405.21060, 2024.

[28] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. Jamba: A hybrid transformer-mamba language model, 2024. URL https://arxiv.org/abs/2403.19887, page 34.

[29] Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? arXiv preprint arXiv:2405.07992, 2024.

[30] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. arXiv preprint arXiv:2401.09417, 2024.

[31] Thomas Roßberg and Michael Schmitt. Estimating ndvi from sentinel-1 sar data using deep learning. In IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium, pages 1412–1415. IEEE, 2022.

[32] Y. Di, Y. Liao, H. Zhou, K. Zhu, Y. Zhang, Q. Duan, et al. Femip: detector-free feature matching for multimodal images with policy gradient. Applied Intelligence, 2023.

[33] Matthew Brown and Sabine Süsstrunk. Multispectral sift for scene category recognition. In CVPR 2011, pages 177–184. IEEE, 2011.

[34] Xue Li, Guo Zhang, Hao Cui, Shasha Hou, Shunyao Wang, Xin Li, Yujia Chen, Zhijiang Li, and Li Zhang. Mcanet: A joint semantic segmentation framework of optical and sar images for land use classification. International Journal of Applied Earth Observation and Geoinformation, 106:102638, 2022.

[35] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In European conference on computer vision, pages 746–760. Springer, 2012.

[36] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017.

[37] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983, 2016.

[38] Yide Di, Yun Liao, Hao Zhou, Kaijun Zhu, Qing Duan, Junhui Liu, and Mingyu Lu. Ufm: Unified feature matching pre-training with multi-modal image assistants. PloS one, 20(3):e0319051, 2025.

[39] Yun Liao, Xuning Wu, Junhui Liu, Peiyu Liu, Zhixuan Pan, and Qing Duan. Fmcfa: a feature matching method for critical feature attention in multimodal images. Scientific Reports, 15(1):6640, 2025.

[40] Yide Di, Yun Liao, Kaijun Zhu, Hao Zhou, Yijia Zhang, Qing Duan, Junhui Liu, and Mingyu Lu. Mivi: Multi-stage feature matching for infrared and visible image. The Visual Computer, 40(3):1839–1851, 2024.

[41] Vassileios Balntas, DP Edgar Riba, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In Proceedings of the British Machine Vision Conference (BMVC), pages 119.1–119.11. BMVA Press, 2016. Available from: https://dx.doi.org/10.5244/C.30.119.

[42] Y. Liao, Y. Di, H. Zhou, A. Li, J. Liu, M. Lu, et al. Feature matching and position matching between optical and sar with local deep feature descriptor. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 15:448–462, 2022.

[43] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3279–3286, 2015.

[44] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In Proceedings of the ieee/cvf conference on computer vision and pattern recognition, pages 8092–8101, 2019.

**Rongrui Teng** received the M.S. degree in Computer Science from Yunnan University, Kunming, China, in June 2025. He is currently a Ph.D. student at Xidian University, Xi'an, China. His research interests include computer vision, multimodal image matching, and deep learning–based feature representation.
https://orcid.org/0009-0002-0990-9432

**Yun Liao** is a Lecturer at the School of Software, Yunnan University, China. He received the BE degree in Communication Engineering from Yunnan University in 2002, the MS degree in software engineering from the Yunnan University in 2009, the PhD degree in Software Engineering from Yunnan University in 2012. His current research interests include machine learning, image processing and deep learning.
https://orcid.org/0000-0002-5555-5236

**Qing Duan** born in 1975, Ph. D., associate researcher at the School of Software, Yunnan University; Key Laboratory in Software Engineering of Yunnan Province, China. She received her PhD degree in Software Engineering from De Montfort University in UK in 2010 respectively. Her current research interests include data mining and knowledge engineering, intelligent information processing, and software engineering.
https://orcid.org/0000-0002-8101-8629

**Wei Wang** received the Ph.D. degree from Yunnan University, Kunming, China, in 2009. He is currently a full professor at the Software School, Yunnan University, China. His research interests focus on software engineering.
https://orcid.org/0000-0001-6562-1594

**Junhui Liu** received the Ph.D. degree in systems analysis and integration from the Yunnan University, in 2009. He is currently a Lecturer with the School of Software, Yunnan University, China. His current research interests include deep learning and domain-specific modelling.
https://orcid.org/0000-0003-3230-5653

**Fangwei Jin** born in 1999, Master, researcher at Guangzhou Laboratory, China. He received his Master's degree from Yunnan University in 2025. His current research interests include image matching and computer vision.
https://orcid.org/0009-0007-8454-7781