Epipolar Shifted Rectangular Window Transformer for Stereo Matching

Ringo S.W. Chu,
SenseTime Research and Tetras.AI

Jinjin Gu
University of Sydney

Jimmy S. Ren
SenseTime Research and Tetras.AI

Abstract

Stereo matching, a technique used to capture depth for 3D applications, has been revolutionized by learning algorithms, particularly in how they construct stereo cost volumes. Each cost volume construction method comes with its unique benefits and limitations. This paper attempts a new approach for constructing such stereo costs, In particular, we propose a Transformer model that learns self and cross attention within local square and rectangle windows, along with epipolar shift to expand the search scope. The resulted matching cost from our Transformer can produce high quality disparity, as evidenced by competitive performance on public datasets. Additionally, our model showcases robustness against our designed stereo attacks.

Introduction

Depth is an integral component in numerous computer vision systems and applications. Stereo systems replaces expensive LiDAR for autonomous vehicles; Augmented reality and Medical Imagaing systems relies on accurate depth for 3D reconstructions. So far researchers have thrived on developing advance methods for stereo matching, particularly focusing on construction and processing of cost volumes for depth estimation.

The advent of new learning algorithms has pioneered traditional stereo approaches. With advances of deep neural nets, the key foundation for successful stereo matching lies on how cost volume is constructed. For instance, DispNet Mayer et al. (2016) creates a correlation cost volume via patch similarity; GCNet Kendall et al. (2017) constructs a cost volume by concatenating image features at various disparity levels; GwCNet Guo et al. (2019) integrate feature concatenation and correlation for a cost volume; Stereo Transformer Li et al. (2021) computes cost metric by series of self and cross attention on epipolar line pixels. However, each of these methods has its strengths and weaknesses. DispNet Mayer et al. (2016) and Stereo Transformer Li et al. (2021) offers efficiency but sacrifices representation learning power; GCNet Kendall et al. (2017) and GwCNet Guo et al. (2019) demonstrates robust performance, but the use of 3D convolutions limits computation efficiency. The main motivation of this paper is to explore a new way for constructing a stereo matching cost volume.

Conversely, Transformer networks have recently demonstrated vast potential in the field of vision. Particularly, we observe the use of Transformer for solving correspondence Jiang et al. (2021) with cross attention. However, stereo matching is more concerned with dense correspondence within small displacement changes. As such we devise a strategy to perform attention within local rectangles windows for our task. To achieve our goal, we extends the self-attention in the Transformer layer to compute self and cross attention from two input features. This newly formulated attention is applied within local square and rectangular windows to learn features and compute the matching score. However, as highlighted in Figure 2a), missing correspondence could occur if the match lies in neighboring window. To address this, We use *epipolar shift* to displace attention windows for the target feature horizontally to the search direction, effectively expanding the search scope. We directly use the attention score from the Transformer with position adjustment to retrieve a full cost volume for disparity regression. Lastly, we add a restoration module to refine loss image details due to low resolution processing. We depict our method in Figure 1.

We perform comprehensive experiments on various popular datasets, including SceneFlow, KITTI, ETH3D, Middlesbury, and MPI Sintel. Our design demonstrates comparable and competitive performance on these benchmarks against other published methods. We deploy adversarial attacks, including wide baselines and visually imbalance, as analogies to evaluate our model's robustness in common practical scenarios. Under such attacks, our approach perform robustly compared to other models. By conducting ablation studies on key components, we validate the efficacy of the design choices implemented in our model. In addition, our visualizations show that our model is capable of generating high-quality disparity maps for difficult regions, as well as for previously unseen domains.

Although our network employs fairly standard components, we tackle stereo matching through a sequence of operations, thereby removing the need for custom learning layers or modules. Contrary to conventional wisdom, we aim to show that basic components can make a good stereo matching model. The following sections will provide a detailed overview of our approach, a discussion on related work, and an outline of our experiments.

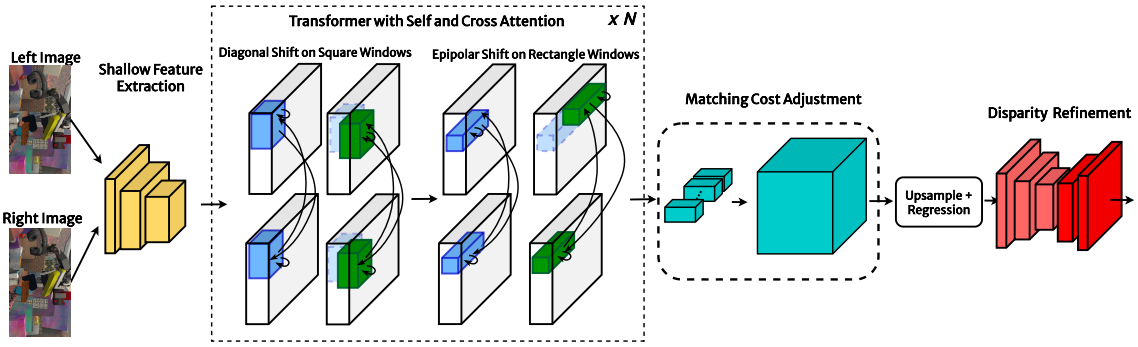


Figure 1: Our overall approach is depicted in the figure and composed of the following steps: ① A weight-sharing backbone is used to obtain a downsampled stereo feature embedding at $\frac{1}{4}$ resolution. ② We stack window self and cross attention Transformer layers, utilizing square and rectangular, diagonal and epipolar shift windows to extract deep features and matching scores. ③ We adjust the dimension of the matching score to retrieve standard cost volume for disparity Regression. ④ Lastly, we use a refinement module to recover lost disparity details.

Method

The goal of rectified stereo matching is to estimate the horizontal offsets for both reference and target images, I^L and I^R. Our approach primarily centers on computing cross-attention for two image inputs within local windows. Each window maintains width size of sw, covering at least the disparity search range D. So that the attention score in Transformer layer measures the corresponding matching pixels along the horizontal scanline.

However, utilizing local windows for correspondence could limit the search scope. For example, the correct correspondence match might be found within adjacent windows. Our model addresses this issue by implementing an epipolar shift at every successive layer. This shift enables the attention window in the target image to move horizontally, thereby facilitating cross-window matching. Our model is illustrated in Figure 1, with each component detailed extensively in the subsequent sections.

Shallow Feature Extraction

Contrary to conventional vision Transformer, we use convolution layers instead of patch embedding and linear projection. This substitution enhances the model's convergence ability Xiao et al. (2021); Wu et al. (2021). In our model, we stack three ConvNeXt Liu et al. (2022) blocks as a shallow feature extractor. This configuration allows us to downsample input images to $x^L, x^R \in \mathbb{R}^{C\times H\times W}$. The backbone weight is shared across both image inputs. We set the output channel $C = 96$, H and W are set to $\frac{1}{4}$ of the input resolution of I^L and I^R.

The Transformer Design

A critical aspect of our task is to produce deep features capable of encapsulating the similarity and mutual dependencies between x^L and x^R. To accommodate our needs, we extend the vanilla attention layer for our Transformer to compute self and cross attention in a single pass.

Window Self and Cross Attention. Assuming we have input features x_l^L, x_l^R, with l denotes the layer index. The features are partitioned into attention windows of size $sh \times sw$. The partitioned feature will then be represented as $\hat{x}_l^L, \hat{x}_l^R \in \mathbb{R}^{\frac{HW}{sh\times sw}\times sh\times sw}$. Following this, we perform multi-head self-attention (MSA) on the concatenated feature $z_l \in \mathbb{R}^{(2\times sh\times sw)\times C}$ from x_l^L and x_l^R. Formally, we express the attention scheme in the following:

$$z_l = [\hat{x}_l^L \parallel \hat{x}_l^R], \tag{1}$$

$$o_l = \mathsf{MSA}(z_l),\ o_l = [o_l^L \parallel o_l^R], \tag{2}$$

Here $\parallel$ signifies the concatenation operation.

Regarding MSA, the input feature z_l will be split into m heads, and thus, the attention h_m for specific m-th head for z_l^m will then can be computed as

$$\mathbf{q}_l^m = z_l^m \mathbf{W}_m^q,\ \mathbf{k}_l^m = z_l^m \mathbf{W}_m^k,\ \mathbf{v}_l^m = z_l^m \mathbf{W}_m^v \tag{3}$$

$$\mathbf{A}_l^m = \mathbf{q}_l^m (\mathbf{k}_l^m)^T, \tag{4}$$

$$\mathbf{h}^m = \mathsf{Softmax}(\frac{\mathbf{A}_l^m}{\sqrt{d}} + \mathbf{B})\mathbf{v}_l^m, \tag{5}$$

where $\mathbf{W}^q, \mathbf{W}^k, \mathbf{W}^v \in \mathbb{R}^{C\times d}$ are the projection metrics for weights, with d as the divided channel dimension. $\mathbf{B}$ is the learnable relative position bias. $\mathbf{A} \in \mathbb{R}^{(2\times sh\times sw)^2}$ is the attention score. The final output o of MSA is the concatenation of all output heads $o = [\mathbf{h}^1 ... \mathbf{h}^m]W^{all}$. Finally, we perform patch merging to retrieve x_{l+1}^L, x_{l+1}^R from attention output.

We claim that our attention scheme can captures the combinations of self and cross attention information from two input features. Moreover, as we compute attentions within local windows, which can effectively mitigate quadratic complexity in standard Transformer.

Attention Score as Matching Score. Considering that stereo matching predominantly focuses on pixel correspondence along the epipolar line. With such motivation, we partition input features x_l^L, x_l^R into rectangular windows, thus allows the capture of horizontal pixel-wise correspondences through the lens of cross-attention. We use such cross-attention score information as our matching score. More formally, we set window width sw to maximum disparity search range D, and sh is defined with a small vertical offset.

Nonetheless, pixel matching follows a monotonic order, implying that a subsequent match $p_{i(j-1)}$ can only position on the left of the previous match p_{ij} on the target image.

So it poses a challenge: the potential correspondence might exist outside its local window, as illustrated in Figure 2a). A naive solution would be to extend the window width sw, but this could result in extra computational demands. We address our concern with with *epipolar shift*, where we display the attention window for x_l^R on their $l+1$ layer in the search direction, as showcased in Figure 2b).

Considering the features $\hat{x}_l^L$ and $\hat{x}_l^R$ that are already partitioned into rectangle sized windows (sh, D). We denote the window position using indices i and j, where i and j are respectively ranges from $[1...\frac{H}{sh}]$ and $[1...\frac{W}{D}]$. When forwarding input features to the Transformer with rectangular windows, specifically for the l-th layer, we perform window attention with $\hat{x}_{(i,j)}^L$ and $\hat{x}_{(i,j)}^R$ with attention score $\mathbf{A_l}$ as the intermediate output. In the subsequent $l+1$ layer, *epipolar shift* shifts the attention window for target view feature $\hat{x}_l^R$ horizontally by $-D$. As a result, the attention $\mathbf{A_{l+1}}$ in this layer is computed from $\hat{x}_{(i,j)}^L$ and $\hat{x}_{(i,j-1)}^R$. Up to this step, we concatenate two attention maps to form $[A_l \| A_{l+1}]$ to access a matching score with an expanded search range. We adopt cyclic-shifting when window index $i = 0$. We compare our shift scheme with diagonal shift Liu et al. (2021) in Figure 2c).

The proposed shift offers the advantage of expanding the disparity search range without the need for large windows. Additionally, we set a small vertical window height to for model to learn spatial context and handle unaligned matches.

Overall Transformer Configuration. Our Transformer network comprises N stages, with each stage containing four Transformer layers. The first two layers equips self and cross attention of square window sized $sh, sw = 8, 8$, with overlapping diagonal shifts in the second layer. The last two layers equips the same attention with rectangular windows sized $sh, sw = 4, 48$[1], incorporating epipolar shifts in the fourth layer. The transformer outputs the concatenated attention score from the last two attention layers in the final stage. Configurations within our network set the MLP expansion ratio to $r = 2.66$, the number of MSA heads to $m = 4$, number of stages $N = 8$. We adopt ReLU as the default choice for the activation function.

Post-Processing

Matching Score Adjustment. At this point, we use the attention from the Transformer, as the matching score for disparity map generation. Upon closer inspection, the attention score is essentially a collection of local relative matching maps that incorporate a combination of self and cross-attention from two feature windows, as shown in Figure 3a. For the sake of simplicity, our model directly uses the 'left-to-right' cross-attention slice expressed the attention score as the matching score for further processing.

Specifically, each local matching scores holds dimensions of $(sh \times sw) \times (2 \times sh \times D)$, as $D = sw$. We apply a $\mathsf{conv}_{1\times1}$ to reduce the extra window height dimension to $(sh \times sw) \times (2 \times D)$. Given that each window contains relative matching information, we adjust the disparity dimension so that each pixel holds a displacement probability vector from its perspective. We set a lower triangle pivot at the intersection point, and roll the disparity vector to the width dimension, as illustrated in Figure 3b. We then de-partition all windows all windows to recover per-pixel unaries matching cost volume $\hat{M} \in \mathbb{R}^{H\times W\times 2D}$ that have been widely deployed in Convolution apporaches, as shown in Figure 3c.

[1]We later set maximum disparity to 192, therefore the downsampled disparity range is calculated as 192/4 = 48

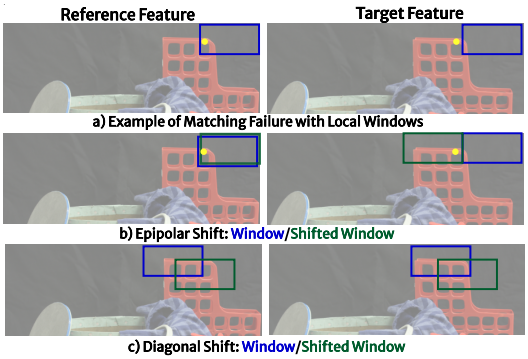


Figure 2: (a) Utilizing local windows could result in unsuccessful match finding, as indicated by the yellow dot marking the correspondence point. (b) Epipolar shift is depicted here. In layer l+1, only the attention windows for the target view feature are displaced horizontally. The boxes are slightly misaligned for the purpose of visualization. (c) The diagonal shift in Swin Transformer Liang et al. (2021). Blue represents the window position in layer l, Green indicates the shifted window position in the successive layer $l+1$.

Upsampling. We use a two-layered convolution to enforce locality coherence for $\hat{M}$. We upsample the cost volume feature to the original resolution using nearest neighbour interpolation, and means to circumvent over-smoothing, often seen with bilinear interpolation. We follow the design principles in Kendall et al. (2017), where we use Softmax and differentiable argmin to regress initial disparity predictions $\mathsf{d}_{\mathsf{init}}$.

Disparity Refinement

Although the transformer can compute a reasonable disparity map, the model is prone to noises due to low-resolution processing, such as depth discontinuities and outliers. To recover fine-grained details, we directly use the Context Adjustment Module described in Li et al. (2021), using the left input image as a reference to compute a refined disparity.

Training Objective

We train our model in an end-to-end supervision manner. Specifically, we supervise all disparity output with Smooth L1 loss, due to its robustness to outlier sensitivity like disparity discontinuities. The loss is given by Equation 6, where $\mathsf{d}_{\mathsf{init}}$ denotes the disparity prediction from Transformer and soft-argmin, and $\mathsf{d}_{\mathsf{refine}}$ denotes the output disparity from disparity refinement module. We set $\lambda_0 = 1.0$ and $\lambda_1 = 1.1$

$$L = \lambda_0 \text{Smooth}_{L1}(d_{\text{init}}, d_{\text{gt}}) + \lambda_1 \text{Smooth}_{L1}(d_{\text{refine}}, d_{\text{gt}}) \quad (6)$$

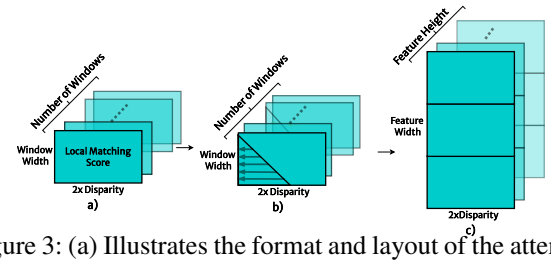


Figure 3: (a) Illustrates the format and layout of the attention map directly acquired from the Transformer. Each window represents a local relative matching score. (b) We roll on the disparity dimension, so each pixel stores a disparity probability vector instead of a relative matching score. (c) The matching score is de-partitioned to retrieve unaries cost volume for disparity regression.

Related Work

Brief History of Stereo Matching

Early Stereo Methods. Early stereo techniques are primarily split into global and local methods. While global methods commonly utilize dynamic programming Hirschmuller (2008); Ohta and Kanade (1985); Forstmann et al. (2004), graph cut Hong and Chen (2004), or generative modeling Geiger, Roser, and Urtasun (2010). Local methods center on processing similarity within local windows, such as Patchmatch in Bleyer, Rhemann, and Rother (2011) or Siamese Networks Zbontar and LeCun (2015); Zagoruyko and Komodakis (2015); Luo, Schwing, and Urtasun (2016); Shaked and Wolf (2017).

Correlation Cost Volume. The advent of deep learning has significantly transformed the field of stereo matching, steering it towards end-to-end methods. Pioneering this shift was DispNetC Mayer et al. (2016), which introduced the first stereo work in this category. The network utilised a correlation layer that computed dot products between two image features from an encoder for cost construction, followed by convolution layers for disparity calculation. Following on the same principle, Pang et al. (2017); Liang et al. (2018) extends Mayer et al. (2016) with a disparity refinement module. Other researches such as Song et al. (2019); Yang et al. (2018) improved upon Mayer et al. (2016) by integrating visual cues into stereo training pipeline. AANet Xu and Zhang (2020) brought forth a novel approach by implementing a multi-scale correlation cost with an innovative aggregation method.

4D Cost Volume. In opposition, GCNet Kendall et al. (2017) adopted a different approach for cost volume construction, building cost volumes by left-right feature concatenation at varying disparity levels, thereby forming 4D cost metrics. Much recent incremental work Chang and Chen (2018); Zhang et al. (2019); Tulyakov, Ivanov, and Fleuret (2018); Khamis et al. (2018); Duggal et al. (2019); Yang et al. (2019); Nie et al. (2019); Xu and Zhang (2020); Zhang et al. (2020) have employed the same cost construction in their model pipeline. GWCNet Guo et al. (2019) and ACVNet Xu et al. (2022) uses a hybrid cost construction encompassing correlation and concatenation Mayer et al. (2016); Kendall et al. (2017) costs through a group-wise correlation volume. Duggal et al. (2019); Shamsafar et al. (2022) address the efficiency concern associated with 3D Convolution. Cheng et al. (2020); Wang et al. (2022) uses AutoML techniques to search for efficient stereo architectures.

Transformer Based Stereo Matching. Among the works closely related to ours is the STTR Transformer Li et al. (2021), which may be seen as a special case of a feature matching Sarlin et al. (2020). STTR employs a Transformer layer to conduct pixel-wise matching along epipolar lines. The model also relies on external mechanisms such as the Sinkhorn algorithm to ensure one-to-one constraints. We contend that pairing all pixels may not be essential, since maximum disparities is proportion to baseline distance, and that affects the area of overlapping region. Thus, it is more advantageous to search correspondences within these areas. Our inspiration is to perform matching in local rectangle windows within a limited extent. Moreover, our method emphasises more on exploiting the modelling ability of Transformer, without the need for external matching modules.

Transformer for Vision Networks

Transformer Vaswani et al. (2017) demonstrated their vast success in language modelling with self-attention layers. In recent years, we see more presences of Transformer in computer vision literatures. Despite the robustness of Transformers, they come with a drawback - their computational complexity escalates quadratically with larger inputs. To address such issue, Swin Transformer Liu et al. (2021) proposes to perform self-attention within local windows, with diagonally shifted scheme to capture global dependencies. Our work can be considered a specific variant of the Swin Transformer, featuring several key adaptations, including rectangular windows, epipolar shifts, and attention score calibration. These simple yet effective changes improve the robustness of Transformers for stereo matching tasks.

Experimental Result

In this section, we verify our approach through three principal experiments:

- Ablation Studies: We dissect and analyse the significance of each component in our model.
- Stereo Adversarial Attacks: We test the model's practical performance and robustness.
- Model Comparisons: We benchmark our model against published methods on popular stereo datasets.

Setup

Dataset. We employs four public datasets for our evaluation. ① SceneFlow Mayer et al. (2016) is a synthetic dataset used for pre-training and ablation experiments. ② KITTI 2015 Menze and Geiger (2015), a real-world dataset with driving scenes, is used for fine-tuning and testing. ③ ETH3D Schöps et al. (2017), an indoor and outdoor grayscale dataset. ④ Middlesbury Scharstein et al. (2014), an indoor scene dataset, are used for testing without fine-tuning. ⑤ MPI Sintel Butler et al. (2012), a synthetic dataset of animated films, is also used for testing without any fine-tuning.

Evaluation Metric. We report our results in the following metrics: End-Point-Error (EPE), D1(%), 1-Pixel(%), 2-Pixel(%), and 3-Pixel(%) errors. The maximum disparity value is set to 192.

Implementation. Our framework is implemented in PyTorch and is available through a GitHub link (hidden for submission). We train our model using AdamW, with an initial learning rate of 0.001 and a weight decay of 0.0001. We employ the OneCycle training scheduler Smith and Topin (2019) for reducing the learning rate. All weights are initialized using Kaiming Initializer. During the training phase, we augment the input images, including random cropping to 288×512, and asymmetric chromatic jittering on brightness ([0.5, 2]), gamma ([0.8, 1.2]) and contrast ([0.8, 1.2]) Yang et al. (2019). We train the network on four Nvidia V100 GPUs with a batch size of 8 image pairs.

Ablation Experiment

We conduct ablations to evaluate the importance of each component within our network. Each component is removed and substituted with an alternative counterpart. Ablation results are presented in Table 1. Unless otherwise specified, the ablated models are trained on the FlyingChair3D for a total of 120,000 steps.

Effectiveness of our Attention Layer. The attention layer plays a crucial role in generating high-quality deep features for matching. In the first setting, we use a Transformer layer devoid of cross-attention, except for the last layer. We also compare our setting with the alternating self and cross attention as in Li et al. (2021); Sun et al. (2021). Our findings indicates that our Transformer can deliver superior performance, trading-off only a minor increase in parameters count. We present the result in Table 1a.

Window Size. The width of the rectangular window size defines the search range for a pixel. To understand their role, we train models with different window sizes to find the optimal choice, while keeping the maximum disparity set to $D = 192$. The results are presented in Table 1b. Decreasing the window width does not lead to stereo collapse, but it prompts a substantial drop in EPE. Conversely, increasing the window width does not necessarily enhance performance. Our findings suggest that a small vertical width generally improves performance, but expanding the window height results in performance degradation. We hypothesize that vertical aspect brings additional spatial context into consideration during matching. When we set windows to 8×48, the computation burden leads to out-of-memory.

Shifting Operations. Our approach employs an epipolar shift to expand the search range. To assess the usefulness our shifting process, we train four models with 4×48 windows to predict two disparity ranges, $D = 96$, $D = 192$, each equipped with either a diagonal Liu et al. (2021) or an epipolar shifting operation. The results are reported in Table 1c. At a lower disparity search range of $D = 96$, the epipolar shift does not contribute, as a local window can cover the whole range. However, upon increasing D to 192, it becomes more apparent that the epipolar shift enhances overall performance.

Positional Encoding. As stereo matching concerns the monotonic ordering property. Positional information contributes how the model learns the matching correspondence. We therefore compare the result of absolute position embedding (APE), learnable relative position bias (LRB), and the combination of both. We find that combining both types of positional encoding doesn't significantly enhance performance; We solely employ LRB in our method as it outperforms others. The results of this comparison are presented in Table 1d.

Stereo Adversarial Attack

A significant barrier to stereo deployment lies in the calibration differences between the actual industrial product and the experimental setup, which can easily result in domain shifts. This section employs adversarial attacks as an analogy to simulate real-world stereo matching scenarios.

Wide Baseline. Commercial applications demand varied stereoscopic configurations, such as larger stereo baselines for self-driving cars versus smaller ones for mobile phones. To simulate large disparities, we adjust the padding of the target-view image. This is achieved by introducing ΔD (extra baseline distance) on the right border. During evaluation, we pad the target image on the right border and apply truncation on the original image boundary. We set values of ΔD as $60, 80, 100, 120$ and report the outcomes in Table 2. We compare our performance with the 3D Conv-based GWCNet Guo et al. (2019) and Transformer-based STTR Li et al. (2021). Our method demonstrates more robustness and suffers less degradation than other methods, except for $\Delta = 120$ due to the limited window size in our design.

Visually Imbalance Setup. Budget-limited smartphones with dual-lens settings often feature a master camera complemented by a less expensive slave camera Liu et al. (2020), resulting in differing resolutions for stereo inputs. To simulate this effect, we downsample the target view by a scale factor s and then upsample to its original resolution via bilinear interpolation. We set s to $1\times$, $2\times$, $3\times$, $5\times$, $8\times$, $12\times$, $15\times$. We report our results in Table 3, where we mark the stereo collapsing point in purple. We show the visualisation result in Figure 4. STTR deteriorates quickly within small s and worsens more severely as s progresses. GWCNet still performs coarse matches at high downgrade levels. Comparatively, our model experiences less degradation and able to acquire general object details in the disparity map across all scale factors.

Quantitative Comparison

In this section, we compare our models with PSMNet Chang and Chen (2018), which utilizes a 3D convolution approach for concatenation-based cost processing; AANetXu and Zhang (2020) an efficient 2D convolution approach for correlation cost processing; GWCNet Guo et al. (2019) is a hybrid approach that combines concatenation and correlation-based cost volumes; CFNet Shen, Dai, and Rao

Transformer Type	EPE	3-Px	D1	Param(M)
w/o Cross	1.35	7.13	5.6	3.76
Alternating	0.92	3.83	2.86	3.76
Ours	**0.86**	**3.73**	**2.65**	3.79

(a) We compare various attention methods in our analysis. Our attention layer demonstrates superior performance compared to alternating self and cross-attention mechanism.

Window Size	EPE	3-Px	Time(s)
1×48	0.98	3.5	0.9
2×24	0.99	3.81	1.0
2×48	0.89	3.74	1.2
2×96	1.05	4.05	1.3
4×24	0.95	3.81	0.9
4×48	**0.86**	**3.73**	1.2
8×24	0.99	5.38	1.0
8×48	OOM	OOM	OOM

(b) Choices for local rectangle window size. 4×48 leads to most optimal performance

Models	D=96		D=192	
	EPE	3-Px	EPE	3-Px
Diagonal Shift	**0.67**	**2.34**	0.95	4.36
Epipolar Shift	0.74	2.63	**0.86**	**3.73**

(c) Our proposed epipolar shifting proves beneficial for larger disparity search range.

	EPE	3-Px	D1
APE	0.90	4.12	**2.59**
LRB	**0.86**	**3.73**	2.65
APE + LRB	0.87	3.86	2.60

(d) Using only the learnable relative position bias (LRB) yields optimal performance than absolute position encoding and combination of both.

Table 1: We present the results of our ablation study, which aims to analyse the significance of each component in our model.

(2021), a 3D convolution approach that focuses on domain-generalisation; STTR Li et al. (2021), a more recent stereo network based on Transformers.

Evaluation on Scene Flow. Our results, presented in Table 4, show that our method surpasses others in terms of EPE, while utilizing fewer parameters than convolution models. Additionally, when comparing our visualization results with Li et al. (2021), as shown in Figure 5. It's evident that our model show advantage in retaining more image details, especially in complex environments.

Evaluation on KITTI. We conduct fine-tuning on the KITTI dataset using a model entirely pre-trained on SceneFlow, allocating 20 out of the 200 images for validation. The benchmark result on the KITTI leaderboard is provided in Table 5. Our performance is inferior to other models, particularly to GWCNet and STTR. We propose that the suboptimal performance may be attributed to the small size of the dataset and the lack of inherent inductive bias typically found in Convolution layers.

One possible way to improve our model performance on KiTTI could be fine-tuning it on a larger dataset, such as Virtual-KITTI Cabon, Murray, and Humenberger (2020). We regard this as future work while remaining a fair comparison with other model setting.

Evaluation on Generalisation Ability. All end-to-end stereo networks are susceptible to domain shifts, potentially leading to performance degradation. We believe that generalisation ability has important practical implication. Thus, we evaluate our model without fine-tuning using the training split of Sintel, Middlesbury, and ETH3D, as well as the KITTI-2015 Guo et al. (2019) training split, all of which represent various degrees of domain shifts. Our generalization results, reported in Table 6, show only minor domain

ΔD	60	80	100	120
GWCNet	3.05	3.26	3.87	**3.99**
STTR	3.50	4.05	4.41	4.99
Ours	**3.02**	**3.19**	**3.60**	4.60

Table 2: EPE performance comparison of stereo algorithms at different extra baseline distances. Our model can handle a reasonably large disparity change when $s < 120$. For the case where $s = 120$, concatenation-based volume suffers from less degradation.

s	1	2	3	5	8	12	15
GWCNET	0.79	0.80	0.82	0.95	1.35	3.12	4.05
STTR	0.50	0.51	**0.52**	**0.57**	0.85	3.67	8.01
Ours	**0.47**	**0.49**	0.54	0.59	**0.74**	**1.34**	**1.75**

Table 3: EPE performance comparison of stereo algorithms under varying imbalance factors s. The purple marker indicates the performance collapse point for the stereo algorithms. Unlike other models, our model continues to perform accurately even when the imbalance factor s is significantly large.

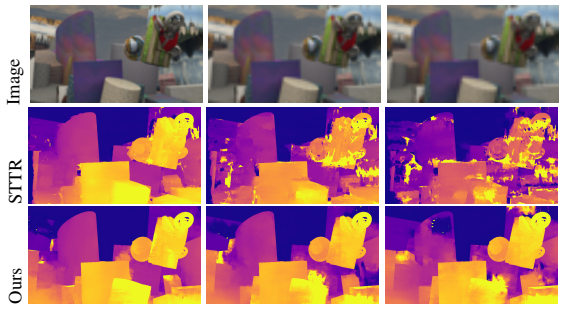
shifts in Sintel, KITTI, and Middlesbury. Our model performs on par or even outperforms other models on these datasets. However, ETH3D, which has the most significant domain gap due to grayscale images with substantial random exposures, shows lower performance without fine-tuning. Our model can withstand and handle minor domain shifts. We provide visual comparisons for Middlesbury Quarter and Sintel datasets in Figure 6 and 7.

Limitation & Discussion

Our model assumes the occluded regions are locally smooth, but occlusions commonly occur in the real world. Therefore method to hand occlusions is an essential task for practical

Model	EPE	3-Px(%)	Param(M)
PSMNET	1.06	4.11	5.2
AANET	0.81	3.32	3.9
GWCNET	0.77	3.30	6.5
CFMet	0.93	4.08	23.1
STTR	0.50	**1.54**	**2.8**
Ours	**0.44**	1.96	3.8

Table 4: Quantitative comparison on Scene Flow test set. **Bold** denotes the best, Underline denotes the second best.



(a) 8× (b) 12× (c) 15×

Figure 4: Stereo matching performance under various imbalance factors $s = 8\times$, $12\times$ and $15\times$. STTR fail at $s \geq 12$. Our model can recover a coarse disparity estimate at an extreme imbalance factor.

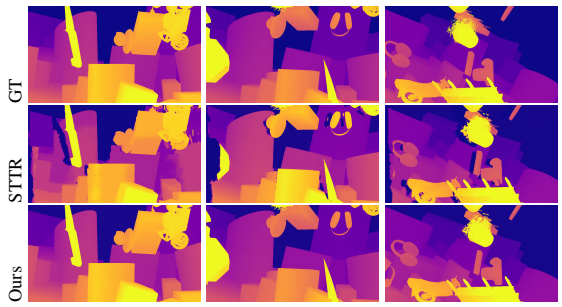


Figure 5: Visualization comparison on SceneFlow.

Figure 6: Generalization comparison on Middlebury Quarter-Resolution dataset. Our model can recover more structural, edge and boundary details.

	All			No Occ		
	d1-bg	d1-fg	d1-all	d1-bg	d1-fg	d1-all
PSMNet	1.86	4.62	2.32	1.71	4.31	2.14
AANet	1.99	5.39	2.55	1.80	4.93	2.32
GWCNet	1.74	3.93	2.11	1.61	3.49	1.92
CFNet	1.54	3.56	1.88	1.43	3.25	1.73
STTR	1.70	3.61	2.01	-	-	-
Ours	1.81	3.92	2.16	1.67	3.53	1.97

Table 5: Results on the KITTI Benchmark leaderboard. We finetune the model on the model pre-trained on SceneFlow. D1 denotes precentage of disparity pixels that are <3px and <5%

Model	KITTI	Sintel	Middlebury		ETH
			Half	Quarter	
PSMNet	1.39	3.31	25.1	14.20	23.80
AANet	1.31	1.89	42.8	35.79	30.44
GWCNet	1.59	1.42	34.20	18.10	30.10
CFNet	1.34	**1.29**	19.5	13.73	5.8
STTR	1.50	3.01	OOM	17.19	17.09
Ours	1.37	1.40	**19.2**	**12.8**	20.57

Table 6: Generalisation results on KITTI-15, MPI Sintel Middlesbury and ETH3D. Models are trained on SceneFlow without fine-tuning. For KITTI-15 and Sintel, we report EPE. For Middlesbury, we report 2-Px. For ETH3D, we report 1-Px

stereo, which is currently missing in our literature. Additionally, Transformers are dependent on large datasets due the the absence of inductive bias. This dependency is more noticeable in our KITTI benchmark, where the Transformer suffered significant degradation due to the limited sample size. New training paradigm or datasets can be used for enhancement when data is not readily available.

Conclusion

We have presented a recipe to perform Stereo Matching using self and cross-attention within rectangular shift windows. Extensive experiments have verified that our performance aligns with that of existing models. It is our hope that our research will provide fresh perspectives and stimulate further exploration within the stereo matching community.

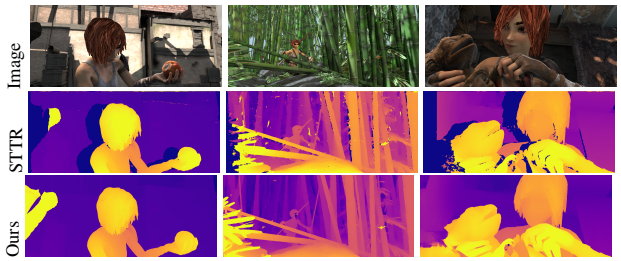


Figure 7: Generalization comparison on MPI Sintel. Best viewed when zoomed in.

References

Bleyer, M.; Rhemann, C.; and Rother, C. 2011. Patchmatch stereo stereo matching with slanted support windows. In *Brit. Mach. Vis. Conf.* 4

Butler, D. J.; Wulff, J.; Stanley, G. B.; and Black, M. J. 2012. A naturalistic open source movie for optical flow evaluation. In Fitzgibbon, A.; Lazebnik, S.; Perona, P.; Sato, Y.; and Schmid, C., eds., *Eur. Conf. Comput. Vis.* 4

Cabon, Y.; Murray, N.; and Humenberger, M. 2020. Virtual kitti 2. 6

Chang, J.-R., and Chen, Y.-S. 2018. Pyramid stereo matching network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 5410–5418. 4, 5

Cheng, X.; Zhong, Y.; Harandi, M.; Dai, Y.; Chang, X.; Li, H.; Drummond, T.; and Ge, Z. 2020. Hierarchical neural architecture search for deep stereo matching. *Adv. Neural Inform. Process. Syst.* 33. 4

Duggal, S.; Wang, S.; Ma, W.-C.; Hu, R.; and Urtasun, R. 2019. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *ICCV*. 4

Forstmann, S.; Kanou, Y.; Ohya, J.; Thuering, S.; and Schmitt, A. 2004. Real-time stereo by using dynamic programming. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 29–29. 4

Geiger, A.; Roser, M.; and Urtasun, R. 2010. Efficient large-scale stereo matching. In *ACCV*. 4

Guo, X.; Yang, K.; Yang, W.; Wang, X.; and Li, H. 2019. Group-wise correlation stereo network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3273–3282. 1, 4, 5, 6

Hirschmuller, H. 2008. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.* 30(2):328–341. 4

Hong, L., and Chen, G. 2004. Segment-based stereo matching using graph cuts. In *IEEE Conf. Comput. Vis. Pattern Recog.* 4

Jiang, W.; Trulls, E.; Hosang, J.; Tagliasacchi, A.; and Yi, K. M. 2021. COTR: Correspondence Transformer for Matching Across Images. In *Int. Conf. Comput. Vis.* 1

Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.; Kennedy, R.; Bachrach, A.; and Bry, A. 2017. End-to-end learning of geometry and context for deep stereo regression. In *Int. Conf. Comput. Vis.* 1, 3, 4

Khamis, S.; Fanello, S.; Rhemann, C.; Kowdle, Adarshand Valentin, J.; and Izadi, S. 2018. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Eur. Conf. Comput. Vis.*, 596–613. 4

Li, Z.; Liu, X.; Drenkow, N.; Ding, A.; Creighton, F. X.; Taylor, R. H.; and Unberath, M. 2021. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *Int. Conf. Comput. Vis.*, 6197–6206. 1, 3, 4, 5, 6

Liang, Z.; Feng, Y.; Guo, Y.; Liu, H.; Chen, W.; Qiao, L.; Zhou, L.; and Zhang, J. 2018. Learning for disparity estimation through feature constancy. In *IEEE Conf. Comput. Vis. Pattern Recog.* 4

Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; and Timofte, R. 2021. Swinir: Image restoration using swin transformer. In *Int. Conf. Comput. Vis.*, 1833–1844. 3

Liu, Y.; Ren, J.; Zhang, J.; Liu, J.; and Lin, M. 2020. Visually imbalanced stereo matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2026–2035. 5

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Int. Conf. Comput. Vis.* 3, 4, 5

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022. A convnet for the 2020s. *IEEE Conf. Comput. Vis. Pattern Recog.* 2

Luo, W.; Schwing, A. G.; and Urtasun, R. 2016. Efficient deep learning for stereo matching. In *IEEE Conf. Comput. Vis. Pattern Recog.* 4

Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.* 1, 4

Menze, M., and Geiger, A. 2015. Object scene flow for autonomous vehicles. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3061–3070. 4

Nie, G.-Y.; Cheng, M.-M.; Liu, Y.; Liang, Z.; Fan, D.-P.; Liu, Y.; and Wang, Y. 2019. Multi-level context ultra-aggregation for stereo matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3278–3286. 4

Ohta, Y., and Kanade, T. 1985. Stereo by intra- and inter-scanline search using dynamic programming. volume PAMI-7, 139–154. 4

Pang, J.; Sun, W.; Ren, J. S.; Yang, C.; and Yan, Q. 2017. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *ICCVW*. 4

Sarlin, P.-E.; DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2020. Superglue: Learning feature matching with graph neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.* 4

Scharstein, D.; Hirschmüller, H.; Kitajima, Y.; Krathwohl, G.; Nešić, N.; Wang, X.; and Westling, P. 2014. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition*. 4

Schöps, T.; Schönberger, J. L.; Galliani, S.; Sattler, T.; Schindler, K.; Pollefeys, M.; and Geiger, A. 2017. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2538–2547. 4

Shaked, A., and Wolf, L. 2017. Improved stereo matching with constant highway networks and reflective confidence learning. In *IEEE Conf. Comput. Vis. Pattern Recog.* 4

Shamsafar, F.; Woerz, S.; Rahim, R.; and Zell, A. 2022. Mobilestereonet: Towards lightweight deep networks for stereo matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2417–2426. 4

Shen, Z.; Dai, Y.; and Rao, Z. 2021. Cfnet: Cascade and fused cost volume for robust stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13906–13915. 5

Smith, L. N., and Topin, N. 2019. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, 369–386. SPIE. 5

Song, X.; Zhao, X.; Fang, L.; and Hu, H. 2019. Edgestereo: An effective multi-task learning network for stereo matching and edge detection. In *Int. J. Comput. Vis.* 4

Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; and Zhou, X. 2021. LoFTR: Detector-free local feature matching with transformers. *IEEE Conf. Comput. Vis. Pattern Recog.* 5

Tulyakov, S.; Ivanov, A.; and Fleuret, F. 2018. Practical deep stereo (pds): Toward applications-friendly deep stereo matching. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc. 4

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is all you need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Adv. Neural Inform. Process. Syst.*, volume 30. 4

Wang, Q.; Shi, S.; Zhao, K.; and Chu, X. 2022. Easnet: Searching elastic and accurate network architecture for stereo matching. In *Eur. Conf. Comput. Vis.* 4

Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; and Zhang, L. 2021. Cvt: Introducing convolutions to vision transformers. *Int. Conf. Comput. Vis.* 2

Xiao, T.; Singh, M.; Mintun, E.; Darrell, T.; Dollar, P.; and Girshick, R. 2021. Early convolutions help transformers see better. In *Adv. Neural Inform. Process. Syst.*, volume 34, 30392–30400. 2

Xu, H., and Zhang, J. 2020. Aanet: Adaptive aggregation network for efficient stereo matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 1959–1968. 4, 5

Xu, G.; Cheng, J.; Guo, P.; and Yang, X. 2022. Attention concatenation volume for accurate and efficient stereo matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 12981–12990. 4

Yang, G.; Zhao, H.; Shi, J.; Deng, Z.; and Jia, J. 2018. Segstereo: Exploiting semantic information for disparity estimation. In *Eur. Conf. Comput. Vis.* 4

Yang, G.; Manela, J.; Happold, M.; and Ramanan, D. 2019. Hierarchical deep stereo matching on high-resolution images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4, 5

Zagoruyko, S., and Komodakis, N. 2015. Learning to compare image patches via convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.* 4

Zbontar, J., and LeCun, Y. 2015. Computing the stereo matching cost with a convolutional neural network. In *IEEE Conf. Comput. Vis. Pattern Recog.* 4

Zhang, F.; Prisacariu, V.; Yang, R.; and Torr, P. H. 2019. Ganet: Guided aggregation net for end-to-end stereo matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 185–194. 4

Zhang, Y.; Chen, Y.; Bai, X.; Yu, S.; Yu, K.; Li, Z.; and Yang, K. 2020. Adaptive unimodal cost volume filtering for deep stereo matching. *AAAI* 12926–12934. 4

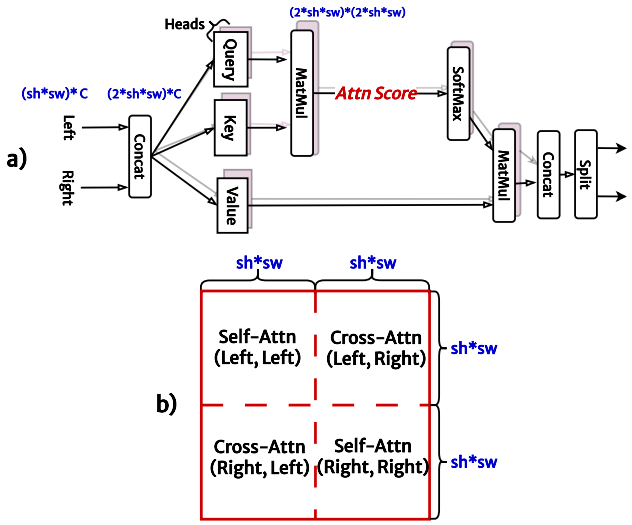


Figure 8: Illustration of our self and cross-attention. (a) illustrates the procedure involved in computing self and cross-attention, while (b) showcases the content of attention encapsulated within the attention score. Here sh and sw denotes window height and width respectively.

Appendix A. More Details about Transformer

Our model primarily relies on the Transformer, and here we provide a more comprehensive explanation of self and cross-attention, with Figure 8 serving as a visual aid. Figure 8a illustrates the flow of attention within a local window. In Figure 8b, we display the distribution of self and cross-attention information within the resulting attention score.

Additionally, we describe the architecture of a stage in Figure 9. Each stage comprises four layers. In the second layer, attention windows for both features shift diagonally. In the fourth layer, only the attention windows of the target view feature are horizontally displaced towards the search direction. Notably, in the final stage, the concatenated attention score is output as the matching score.

Appendix B. Feature Visualisation and Interpretation

As previously noted, robust matching primarily hinges on a strong feature representation ability, especially in the context of stereo challenges, such as textureless regions. To gain the most intuitive insight into what the Transformer learns, we apply PCA reduction from SCI-KIT LEARN to the output feature from the final Transformer layer. The results are presented in Figure 10. This visualization indicates that our layers can extract significant semantic cues, such as object boundaries and edges, and can differentiate between foreground and background areas. These cues embody the necessary geometric knowledge for performing matching in ill-posed regions.

Appendix C. Additional Visual Results

We provide further qualitative results of our model and STTR on SceneFlow, KITTI-2015, MPI Sintel, and

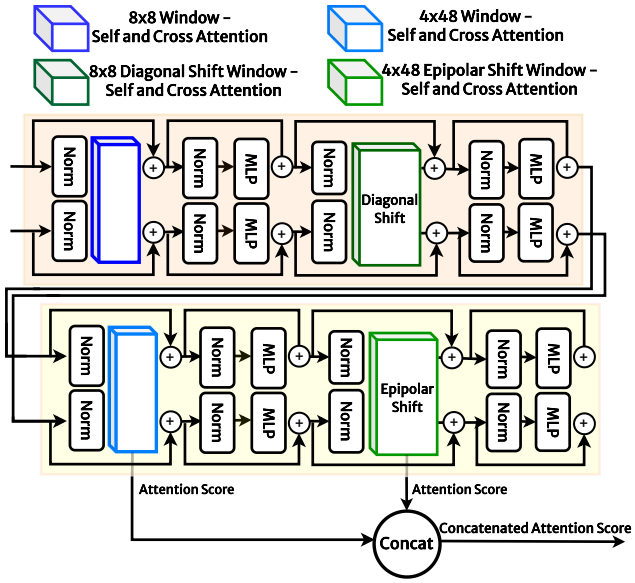


Figure 9: Depiction of a stage in our Transformer. Each stage incorporates four Transformer layers. In the first two layers (depicted in blue), the Transformer employs 8×8 square window partitioning with overlapping diagonal shifting. In the final two layers (illustrated in green), the Transformer utilizes 4×48 rectangular window partitioning with epipolar shifting.

Middlesbury-Quarter.

SceneFlow Additional visual comparisons are presented in Figure 11. All models are pre-trained on Scene Flow.

KITTI-2015. Visualization results for the KITTI-2015 test set are displayed in Figure 12. All models are initially trained on Scene Flow and subsequently fine-tuned on KITTI 2015. Our model is somewhat less effective for this dataset. Given that KITTI comprises a smaller dataset with only sparse disparity point clouds for non-distant areas. As such, the training process becomes challenging, and that the Transformer struggles to generalise. Consequently, artifacts are noticeable in regions where ground truth disparities are absent during training.

Middlesbury. Additional quantitative results on samples from the Middlesbury Quarter Resolution Dataset are provided in Figure 13. All models are trained on Scene Flow without any fine-tuning.

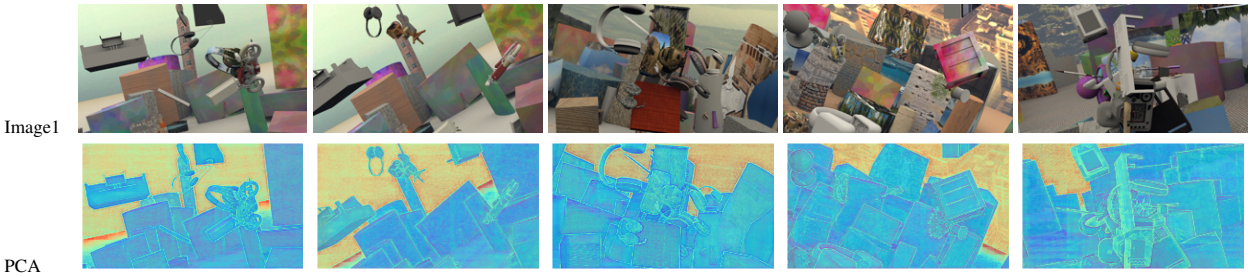


Figure 10: PCA visualization results. Our Transformer is capable of extracting features with meaningful semantic representations essential for stereo matching.

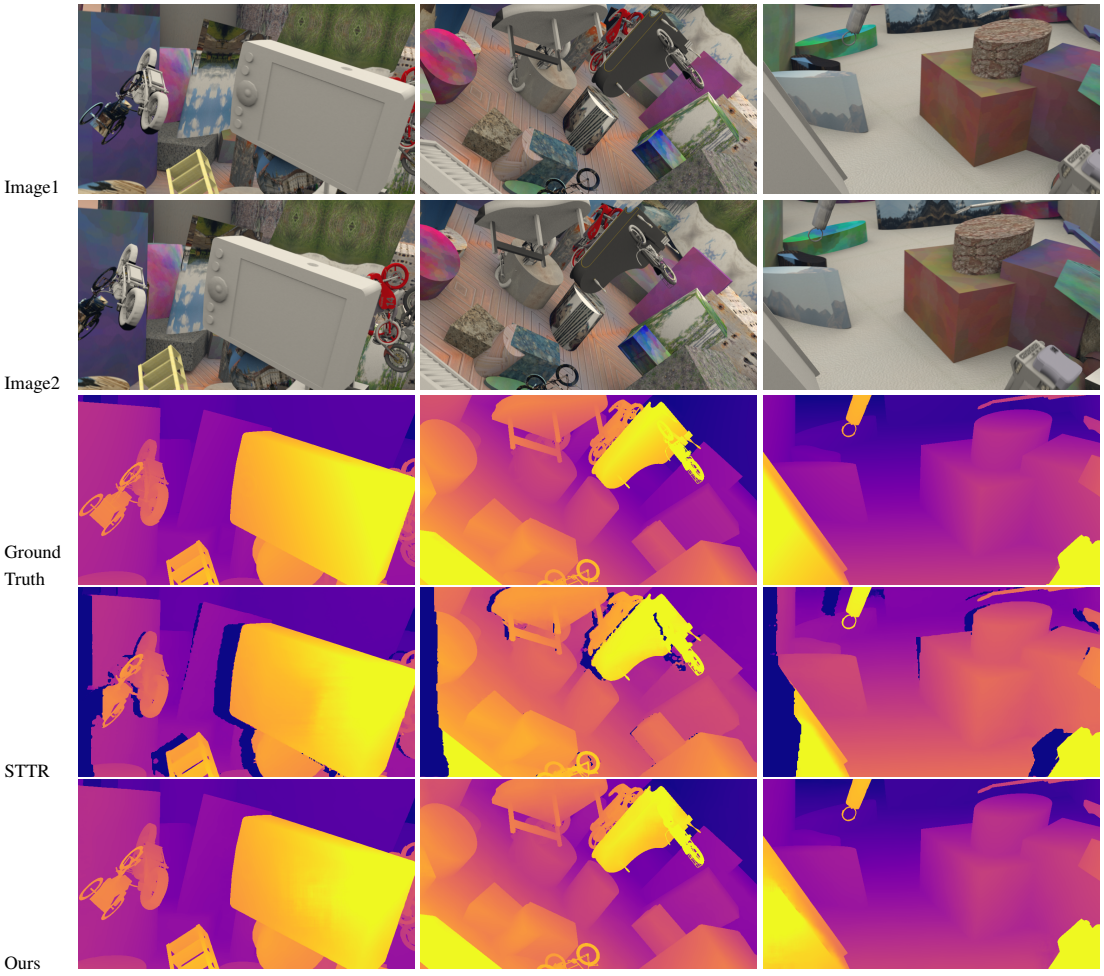


Figure 11: More visual comparison on Scene Flow.

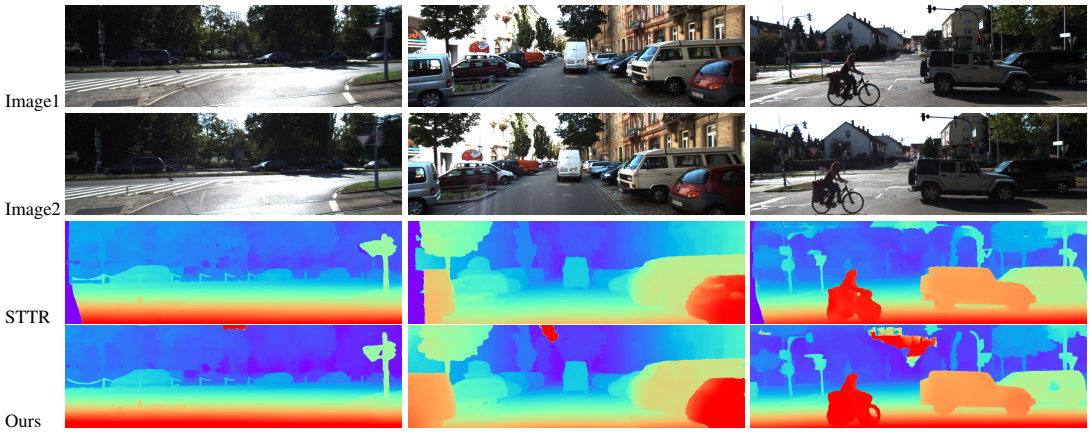


Figure 12: Visual comparison on KITTI 2015 test split. Noticeable artifacts can be observed in regions where disparities are absent during training.

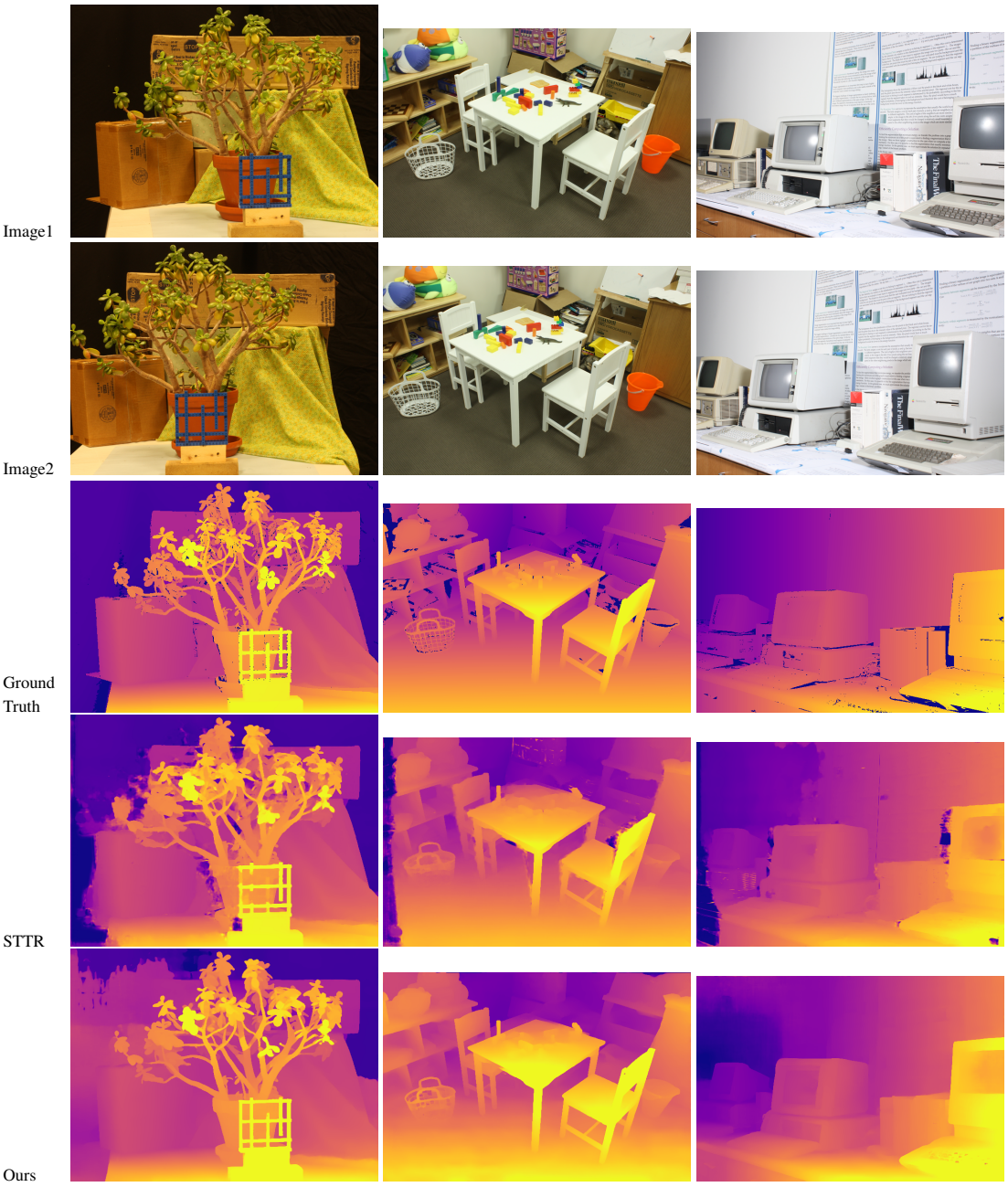


Figure 13: More generalisation visual results on Middlesbury Quarter Resolution Dataset.