LoS: Local Structure-guided Stereo Matching

Kunhong Li[1,2] Longguang Wang[4] Ye Zhang[1,2] Kaiwen Xue[5] Shunbo Zhou[5] Yulan Guo[3*]
[1]Sun Yat-Sen University [2]The Shenzhen Campus of Sun Yat-Sen University [3]National University of Defense Technology
[4]Aviation University of Air Force [5]Huawei Cloud Computing Technologies Co., Ltd.

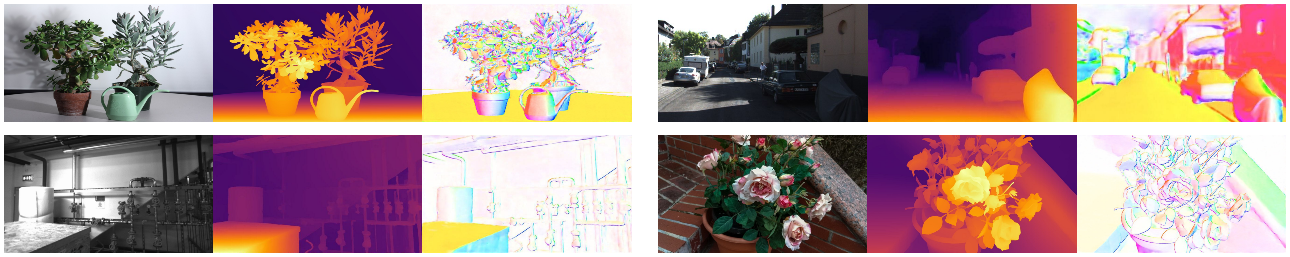


Figure 1. Example results of LoS on Middlebury, ETH3D, KITTI2012 and Holopix 50k. In each example, we show the left image, estimated disparity map and slant plane. Note that, these are all unseen samples for LoS.

Abstract

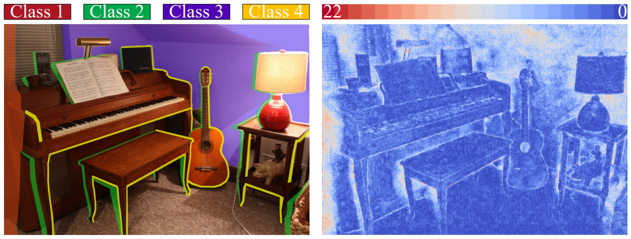
Estimating disparities in challenging areas is difficult and limits the performance of stereo matching models. In this paper, we exploit local structure information (LSI) to better handle these areas. Specifically, our LSI comprises a series of key elements, including the slant plane (parameterised by disparity gradients), disparity offset details and neighbouring relations. This LSI empowers our method to effectively handle intricate structures, including object boundaries and curved surfaces. We bootstrap the LSI from monocular depth and subsequently refine it to better capture the underlying scene geometry constraints in an iterative manner. Building upon the LSI, we introduce the Local Structure-Guided Propagation (LSGP), which enhances the disparity initialization, optimization, and refinement processes. By combining LSGP with a Gated Recurrent Unit (GRU), we present our novel stereo matching method, referred to as ***Lo****cal* ***S****tructure-guided stereo matching (LoS). Remarkably, LoS achieves top-ranking results on four widely recognized public benchmark datasets (ETH3D, Middlebury, KITTI 15 & 12) and robust vision challenge, demonstrating the superior capabilities of our model.*

1. Introduction

The primary objective of stereo matching is to identify accurate correspondences, referred to as disparities, between pairs of input images. Existing learning-based methods commonly regress the disparity map based on the raw feature correlations/costs. Specifically, most previous methods adopt filtering-based techniques to refine the cost volume and then regress accurate disparities. These methods first construct a 3D/4D cost volume and then filter this volume using 2D/3D convolutional neural networks (CNNs). However, these methods require a pre-defined disparity range to produce satisfactory results. To remedy this limitation, another group of methods [18, 20, 46, 51] leverage optimization-based technique to optimize disparities directly using a 2D convolutional gated recurrent unit (ConvGRU) without relying on pre-defined disparity ranges. Nonetheless, these methods usually require a substantial number of iterations to achieve convergence.

Challenging areas in stereo pairs are the primary reason why optimization-based methods necessitate numerous iterations to yield satisfactory results. As shown in Fig. 2, these challenging areas in the left image of a stereo pair encompass: Class 1, regions on the left side that are outside the visible range of the right view. Class 2, occluded areas on the left side of foreground objects. Class 3, textureless areas. Class 4, edge areas (due to blurry edges and image downsampling [4, 51]). For these challenging areas, accurate disparities cannot be obtained solely from the appearance information in these areas due to the ambiguity of pixel correspondence. Instead, geometry and depth cues in a wider neighboring areas should be adopted to reason accurate disparities. To this end, several efforts have been made to formulate local geometry as a slant plane[4, 14, 35, 36, 42], which is represented using either

[*] Corresponding author: Yulan Guo (yulan.guo@nudt.edu.cn).



(a) The left image of a stereo pair

(b) The number of iterations required for convergence

Figure 2. An illustration of four categories of challenging areas, including left-side areas that are out of right view's visible range (class 1), occluded areas (class 2), textureless areas (class 3), and edge areas (class 4). The number of iterations to achieve convergence that required for RAFT-Stereo [20] is shown in (b).

the plane's normal vector [4, 14, 35, 42] or the disparity gradient [36]. However, the slant plane is limited to simple planar structures and cannot well model non-planar geometries, such as object boundaries and curved surfaces.

In this paper, we propose a novel representation of local structural information (LSI) by combining the slant plane with disparity offset details and neighbor relations. Our proposed LSI explicitly characterizes local structures and exhibits improved performance in modeling non-planar structures. On top of this LSI representation, we introduce a local structure-guided propagation (LSGP) method to optimize the disparity map by propagating low-uncertainty disparities while updating high-uncertainty ones. Integrating our approach into an optimization-based framework, we present a stereo matching method called LoS, which stands for **Lo**cal **S**tructure-guided stereo matching.

Our primary contributions can be summarized as:

- We propose LoS, a local structure-guided stereo matching method that integrates structure information to improve stereo matching performance in challenging areas.
- We introduce local structure-guided propagation (LSGP) to explicitly leverage structure information for updating disparities in challenging areas.
- Extensive experiments on four popular public benchmarks demonstrate the effectiveness of LoS.

2. Related Work

In this section, we first review stereo matching methods using filtering-based and optimization-based techniques for disparity regression. Then, we discuss recent methods that exploits local structure information for disparity estimation.

2.1. Filtering-based Methods

Filtering-based methods commonly employ 2D/3D CNNs to process 3D/4D cost volumes for disparity regression. Following the traditional pipeline, Zbontar and LeCun [49] replace the hand-crafted cost with learned matching score, and regularize it with semi-global matching (SGM) [15]. DispNet [24] is the first end-to-end stereo matching method that introduces the explicit correlation computation into disparity estimation. To boost the performance of stereo matching, 4D cost volumes are constructed to represent the scene geometry. The cost volume can be built with concatenated features [8, 17] or group-wise correlation [12] on a fixed scale [8, 12, 17, 19, 26] or multiple levels [22, 31, 32]. Since these methods achieve cost aggregation/regularization with 3D CNNs, they usually relies on a preset disparity range to reduce memory cost. To further reduce computational and memory costs, commonly used the techniques include compressing the channel dimension of cost volume and aggregating the cost with 2D convolution [47], and enforcing limit on the size of 4D cost volume [10, 13, 44]. However, these methods usually suffer from performance drop. Additionally, filtering-based methods are usually poor in generalization [32, 50].

2.2. Optimizing-based Methods

Optimization-based methods iteratively update the disparity in an optimization-based framework. Specifically, GRU-based optimization methods [37, 40, 48] show great power in dense correspondence matching tasks. Inspired by [37], Lipson *et al*. propose RAFT-Stereo [20], which iteratively optimizes the disparity map with multi-level ConvGRU and multi-level cost volumes. Li *et al*. propose CREStereo [18] by introducing a cascade optimization architecture and an adaptive sampling strategy to enhance model performance in practical applications. Zhao *et al*. propose DLNR [51] to hold the detail information in feature maps using a decoupled Long Short-Term Memory (LSTM). Although these optimization methods achieve remarkable performance, they need dozens of iterations to achieve convergence. To alleviate this problem, Xu *et al*. [46] propose IGEV to process the cost volume using a light-weighted 3D CNN as the structure information for GRU updating. However, due to the heavy memory cost and poor generalization ability of 3D CNN, the structure information provided by the cost volume is limited by the disparity range and sometimes noisy (containing artifacts).

2.3. Stereo Matching with Local Structure

Slant plane is widely applied in stereo matching [4, 6, 23, 35]. The plane parameters, either single-scale [4, 33] or multi-scale [6], are usually random initialized [4, 14, 35] or initialized from sparse matching [33], and are usually iteratively optimized [4, 6, 14, 33, 35]. Bleyer *et al*. [4] propose PatchMatchStereo, a method that employs random initialization of normal vectors and refines them through patchmatch propagation. Chakrabarti *et al*. [7] propose CoRStereo based on multi-level slant plane represented by normal vectors and optimized by a consensus framework. Recently, the slant plane is introduced into deep-

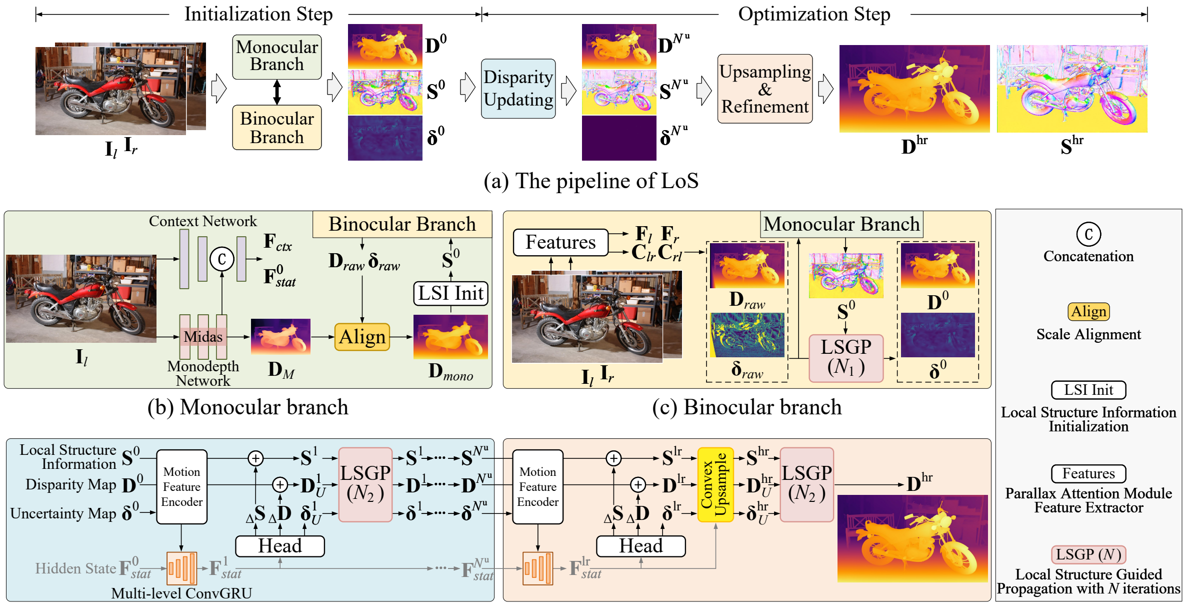


Figure 3. The architecture of LoS. We show the overall pipeline in (a), and illustrate the details of monocular branch (b), binocular branch (c), disparity updating (d) and upsampling and refinement (e). Note that in (d) and (e), we omit the variables $\mathbf{F}_l$, $\mathbf{F}_r$ and $\mathbf{F}_{ctx}$, which are used but not updated by GRU and LSGP, and we simplify the $\mathbf{S}^i$ updating procedure, which is detailed in steps 5 and 6 of Alg.1.

learning based methods. Wang *et al.* [42] introduce the slant plane into correlation sampling and disparity refinement with learned plane parameters. Tankovich *et al.* [36] use the tile representation, which contains several pixels and a slant plane represented by disparity gradient, to efficiently propagate information and achieve accurate disparity and local structure estimation.

3. Methodology

Our LoS consists of an initialization step and an optimization step, with the core module being the LSGP, as illustrated in Fig. 3(a). Given a stereo pair $\{\mathbf{I}_l, \mathbf{I}_r\}$ with dimensions $sH \times sW$, where $s = 4$ is the spatial scale factor. The initialization step initializes the disparity map $\mathbf{D}^0$ and local structure information $\mathbf{S}^0$ for the optimization step. The optimization step iteratively updates $\mathbf{D}^0$ and $\mathbf{S}^0$ for N iterations, resulting in $\mathbf{D}^N$ and $\mathbf{S}^N$, which are then upsampled and refined to obtain the final disparity map $\mathbf{D}^{\text{hr}}$ with a resolution of $sH \times sW$.

3.1. Local Structure-guided Propagation

The objective of our LSGP module is to propagate low-uncertainty disparities to update high-uncertainty disparities under the guidance of local structure information (LSI). Compared to the previous GRU-based propatation technique, our approach eliminates time-consuming operations (such as feature warping for correlation calculation, correlation sampling from volumes), making our propagation significantly more efficient, see Fig. 5(b).

Local Structure Information (LSI). For a pixel at location $\mathbf{p} = (h, w)$, its neighboring window is defined as a 3×3 square centered at $\mathbf{p}$: $\mathcal{N}(\mathbf{p}) = \{\mathbf{p}_i | \mathbf{p}_i = (h_i, w_i)\}$, where $\mathbf{p}_i$ is constrained with $|h_i - h| \leq 1$, $|w_i - w| \leq 1$. Then, we use the LSI $\mathbf{S}(\mathbf{p})$ to describe the local structure of $\mathcal{N}(\mathbf{p})$. The LSI $\mathbf{S}$ is based on the slant plane, which is represented by disparity gradients $\mathbf{G}$. Since $\mathbf{G}$ is only limited to planar structures, we further introduce disparity offset details $\mathbf{O}$ and local relations $\mathbf{R}$ to model non-planar structures such as edges and curve surfaces. Therefore, the LSI is defined as $\mathbf{S} = \{\mathbf{G}, \mathbf{O}, \mathbf{R}\}$. Here, $\mathbf{G}$ with dimensions $H^{\text{p}} \times W^{\text{p}} \times 2$ describes the horizontal and vertical gradients of disparities, while $\mathbf{O}$ and $\mathbf{R}$ are with dimensions $H^{\text{p}} \times W^{\text{p}} \times 9$. The spatial dimension $(H^{\text{p}}, W^{\text{p}}) \in \{(H, W), (sH, sW)\}$.

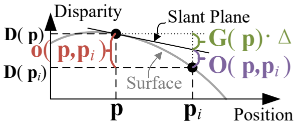


Figure 4. 1d illustration for Eq. 2.

Propagation Process. Given a disparity map $\mathbf{D}$, an uncertainty map $\boldsymbol{\delta}$ and LSI $\mathbf{S} = \{\mathbf{G}, \mathbf{O}, \mathbf{R}\}$, we propagate neighboring disparities $\mathbf{D}(\mathcal{N}(\mathbf{p}))$ and uncertainties $\boldsymbol{\delta}(\mathcal{N}(\mathbf{p}))$ to update $\mathbf{D}(\mathbf{p})$ and $\boldsymbol{\delta}(\mathbf{p})$ according to $\mathbf{S}(\mathbf{p})$:

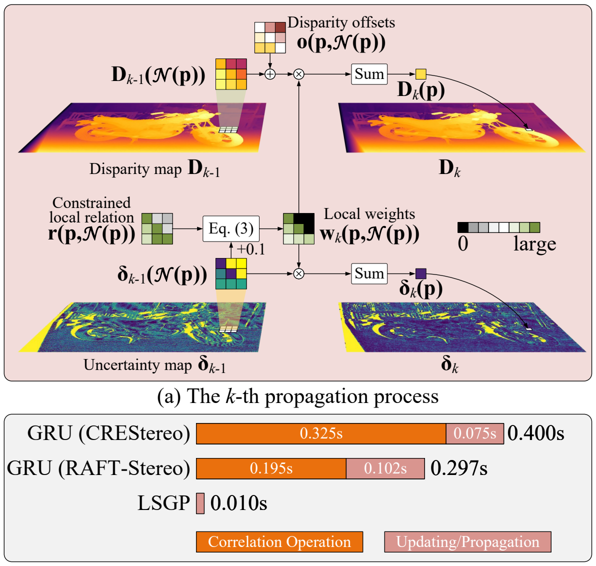
$$\mathbf{D}_k(\mathbf{p}) = \sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} \mathbf{w}_k(\mathbf{p}, \mathbf{p}_i)(\mathbf{D}_{k-1}(\mathbf{p}_i) + \mathbf{o}(\mathbf{p}, \mathbf{p}_i)),$$

$$\boldsymbol{\delta}_k(\mathbf{p}) = \sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} \mathbf{w}_k(\mathbf{p}, \mathbf{p}_i)\boldsymbol{\delta}_{k-1}(\mathbf{p}_i), \tag{1}$$

the subscript k denotes the k-th iteration, and the lowercase $\mathbf{o}$ is the disparity offsets derived from $\mathbf{G}$ and $\mathbf{O}$:

$$\mathbf{o}(\mathbf{p}, \mathbf{p}_i) = \mathbf{G}(\mathbf{p}) \cdot \Delta\mathbf{p}_i + \mathbf{O}(\mathbf{p}, \mathbf{p}_i), \tag{2}$$

where $\Delta\mathbf{p}_i = \mathbf{p} - \mathbf{p}_i$ is the pixel offset vector. $\mathbf{o}$ is computed only once for each propagation process. Besides, $\mathbf{w}_k$



(a) The k-th propagation process

(b) Time consumption of 50 updating iterations

Figure 5. Local Structure-Guided Propagation. We illustrate the k-th propagation process in (a), and compare the time consumption of GRU updating and LSGP on images with 960 × 640 resolution for 50 iterations in (b).

is the weights and $\sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} \mathbf{w}_k(\mathbf{p}, \mathbf{p}_i) = 1$. To suppress the spreading of high-uncertainty disparities, we split $\mathcal{N}(\mathbf{p})$ into $\mathcal{N}^+(\mathbf{p})$ and $\mathcal{N}^-(\mathbf{p})$ according to $\boldsymbol{\delta}_{k-1}$. $\mathcal{N}^+(\mathbf{p})$ contains all the low-uncertainty neighbours $\mathbf{p}_i$ of $\mathbf{p}$, where low uncertainty is defined as $\boldsymbol{\delta}_{k-1}(\mathbf{p}_i) \leq \boldsymbol{\delta}_{k-1}(\mathbf{p}) + \delta$ with the margin $\delta = 0.1$. Note that during training, we set $\mathcal{N}^+(\mathbf{p}) = \mathcal{N}(\mathbf{p})$ and $\mathcal{N}^-(\mathbf{p}) = \emptyset$ to enhance model robustness. $\mathbf{w}_k$ is computed according to this split:

$$\mathbf{w}_k(\mathbf{p}, \mathbf{p}_i) = \begin{cases} \frac{\exp(\hat{\boldsymbol{\delta}}_{k-1}(\mathbf{p}_i)\mathbf{r}(\mathbf{p},\mathbf{p}_i))}{\sum\limits_{\mathbf{p}_j \in \mathcal{N}^+} \exp(\hat{\boldsymbol{\delta}}_{k-1}(\mathbf{p}_j)\mathbf{r}(\mathbf{p},\mathbf{p}_j))} & , \text{ if } \mathbf{p}_i \in \mathcal{N}^+ \\ 0 & , \text{ if } \mathbf{p}_i \in \mathcal{N}^- \end{cases}, \tag{3}$$

where $\hat{\boldsymbol{\delta}}_{k-1}$ and $\mathbf{r}$ are derived from uncertainty map $\boldsymbol{\delta}_{k-1}$ and local relations $\mathbf{R}$. To achieve numerical stability and ensure that $\boldsymbol{\delta}$ always takes effect, all elements in $\mathbf{r}$ are constrained to be smaller than -1, *i.e.* $\mathbf{r}(\mathbf{p}, \mathbf{p}_i) = \mathbf{R}(\mathbf{p}, \mathbf{p}_i) - \max(\mathbf{R}(\mathbf{p})) - 1$. Since the elements of $\mathbf{r}$ are always smaller than -1, we use uncertainty instead of confidence as the scaling factor. To ensure that $\mathbf{R}$ always takes effect, we use $\hat{\boldsymbol{\delta}}_{k-1} = \boldsymbol{\delta}_{k-1} + 0.1$.

Uncertainty vs. Local Relations. Uncertainties and local relations play different roles in Eq. 3. The uncertainty $\boldsymbol{\delta}(\mathcal{N}(\mathbf{p}))$ is the state of a pixel to indicate how reliable the current disparity value is, so it should keep the same updating tracks as the disparity value. Therefore, $\boldsymbol{\delta}(\mathcal{N}(\mathbf{p}))$ is updated with the same weight as $\mathbf{D}(\mathcal{N}(\mathbf{p}))$ in Eq. 1. While the local relation $\mathbf{R}(\mathbf{p}, \mathbf{p}_i)$ is the inherent correlation between the pixel pairs, and should remain unchanged in LSGP.

Model Architecture. There is no learnable parameters in LSGP, all variables are updated by GRU. To adapt LSGP, we slightly modify the multi-level GRU used in RAFT-Stereo [20]: 1) We expand the output channel of the disparity head to 13, allocating 1 channel for disparity redisual $_\triangle\mathbf{D}$, 1 for uncertainty $\boldsymbol{\delta}_U^i$, 2 for disparity gradient residual $_\triangle\mathbf{G}$ and 9 for disparity offset details residual $_\triangle\mathbf{O}$ (refer to step 3 in Alg. 1). 2) We introduce a two-layer head Φ to initialize and update local relations $\mathbf{R}$.

3.2. Initialization Step

The initialization step consists of two branches. The binocular branch initially passes raw disparity map $\mathbf{D}_{raw}$ and uncertainty map $\boldsymbol{\delta}_{raw}$ to the monocular branch, and then the monocular branch initialize the LSI $\mathbf{S}^0$ and feeds it back to the binocular branch.

3.2.1 Monocular Branch

In the monocular branch, $\mathbf{I}_l$ is encoded into a context feature map $\mathbf{F}_{ctx}$ and hidden state $\mathbf{F}^0_{stat}$ using a context network. Besides, we also introduce the depth prior by estimating the monocular depth of $\mathbf{I}_l$. The monocular depth network is an off-the-shelf model, MiDaS [3, 28], with a fixed I/O size of 384 × 384. The monocular depth generated by MiDaS is then aligned to the true scale and upsampled to create the disparity map $\mathbf{D}_{mono}$ with a spatial dimension of $H \times W$.

Scale Alignment. MiDaS outputs an up-to-scale depth map $\mathbf{D}_M$, where each pixel's depth is represented as inverse depth, akin to virtual disparity. To align monocular depth $\mathbf{D}_M$ with raw disparity $\mathbf{D}_{raw}$, we calculate scale and shift factors by solving a weighted least square problem:

$$\begin{aligned} s_d, t_d &= \arg\min_{s_d, t_d} \sum_{i=1}^{384*384} O(i), \\ O(i) &= (1 - \boldsymbol{\delta}_{raw}(i))(s_d \mathbf{D}_M(i) + t_d - \mathbf{D}_{raw}(i))^2. \end{aligned} \tag{4}$$

Both $\mathbf{D}_{raw}$ and $\boldsymbol{\delta}_{raw}$ are resampled to 384 × 384 to match the size of $\mathbf{D}_M$.

Finally, $\mathbf{D}_{mono}$ is computed and upsampled to the size of $H \times W$: $\mathbf{D}_{mono} = \text{Bilinear Upsample}(s_d \mathbf{D}_M + t_d)$. Although $\mathbf{D}_{mono}$ is aligned to the estimated true scale, we do not use it as the initial disparity map directly due to the lacks of object details. Instead, we use $\mathbf{D}_{mono}$ as a prior to initialize $\mathbf{S}^0$ and then optimize $\mathbf{D}_{raw}$ with LSGP.

LSI Initialization. We first initialize the local relations $\mathbf{R}^0 = \Phi(\text{cat}(\mathbf{D}_{raw}, \mathbf{D}_{\text{mono}}, \mathbf{F}_{\text{ctx}}, \mathbf{F}^0_{\text{stat}}))$, where cat() represents the concatenation operator. Subsequently, we have $\mathbf{o}^0(\mathbf{p}, \mathbf{p}_i) = \mathbf{D}_{mono}(\mathbf{p}) - \mathbf{D}_{mono}(\mathbf{p}_i)$, and initialize disparity gradients $\mathbf{G}^0$ and disparity offset details $\mathbf{O}^0$:

$$\begin{aligned} &\mathbf{G}^0(\mathbf{p}) = \arg\min_{\mathbf{G}(\mathbf{p})} \sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} \mathbf{w}(\mathbf{p}, \mathbf{p}_i)(\mathbf{G}(\mathbf{p}) \cdot \Delta\mathbf{p}_i - \mathbf{o}^0(\mathbf{p}, \mathbf{p}_i))^2, \\ &\mathbf{O}^0(\mathbf{p}, \mathbf{p}_i) = \mathbf{o}^0(\mathbf{p}, \mathbf{p}_i) - \mathbf{G}^0(\mathbf{p}) \cdot \Delta\mathbf{p}_i, \end{aligned} \tag{5}$$

where $\mathbf{w}(\mathbf{p}, \mathbf{p}_i)$ is computed from Eq. 3 based on $\boldsymbol{\delta}_{raw}$, $\mathbf{R}^0$ and $\mathcal{N}^+(\mathbf{p}) = \mathcal{N}(\mathbf{p})$. And we have $\mathbf{S}^0 = \{\mathbf{G}^0, \mathbf{O}^0, \mathbf{R}^0\}$.

3.2.2 Binocular Branch

In the binocular branch, $\mathbf{I}_l$ and $\mathbf{I}_r$ are encoded into $\mathbf{F}_l$, $\mathbf{F}_r$, $\mathbf{C}_{lr}$, $\mathbf{C}_{rl}$, $\mathbf{D}_{raw}$, and $\boldsymbol{\delta}_{raw}$ using a Parallax Attention Mechanism (PAM)-based feature extractor [41]. Subsequently, $\mathbf{D}_{raw}$ and $\boldsymbol{\delta}_{raw}$ are optimized as $\mathbf{D}^0$ and $\boldsymbol{\delta}^0$ with LSGP.

PAM Extractor. In the PAM extractor, the input images are first progressively downsampled to 1/32 resolution and then gradually upsampled to 1/4 resolution using a Res-UNet. The feature pyramids from the Res-UNet are then fed into four parallax attention modules to produce the correspondence feature maps $\mathbf{F}_l$ and $\mathbf{F}_r$ and two correlation matrices: $\mathbf{C}_{lr}$ with dimensions $H \times W_l \times W_r$ and $\mathbf{C}_{rl}$ with dimensions $H \times W_r \times W_l$. Note that both $\mathbf{C}_{lr}$ and $\mathbf{C}_{rl}$ are softmaxed along the last dimension. Finally, $\mathbf{D}_{raw}$ is regressed from $\mathbf{C}_{lr}$ and $\boldsymbol{\delta}_{raw}$ is estimated from the left-right consistency between $\mathbf{C}_{lr}$ and $\mathbf{C}_{rl}$:

$$\begin{aligned} \mathbf{D}_{raw}(h, w_l) &= \sum_{w_r=0}^{w_l} (w_l - w_r)\mathbf{C}_{lr}(h, w_l, w_r), \\ \boldsymbol{\delta}_{raw}(h, w_l) &= \sum_{w_r=0}^{w_l} \mathbf{C}_{lr}(h, w_l, w_r)\mathbf{C}_{rl}(h, w_r, w_l). \end{aligned} \tag{6}$$

LSGP. LSGP is applied on resolution $(H^{\mathrm{p}}, W^{\mathrm{p}}) = (H, W)$ with N_1 iterations, as shown in Fig 3(c). Since the disparity map $\mathbf{D}_{raw}$ are extremely noisy, N_1 is relatively large to ensure sufficient propagation.

3.3. Optimization Step

3.3.1 Disparity Updating

Disparity updating contains N iterations. For the i-th iteration, previous disparity map $\mathbf{D}^{i-1}$, uncertainty map $\boldsymbol{\delta}^{i-1}$ and LSI $\mathbf{S}^{i-1}$ are first updated by a multi-level ConvGRU to obtain $\mathbf{D}^i_U$, $\boldsymbol{\delta}^i_U$ and $\mathbf{S}^i$. Subsequently, $\mathbf{D}^i_U$ and $\boldsymbol{\delta}^i_U$ are further updated by LSGP based on $\mathbf{S}^i$. The process of a single iteration is illustrated in Fig. 3(d) and Alg. 1.

Algorithm 1 A Single Iteration for Disparity Updating

```
$\mathbf{F}^{i-1}_{mot} \leftarrow \mathrm{MFE}(\mathbf{F}_l, \mathbf{F}_r, \mathbf{D}^{i-1}, \boldsymbol{\delta}^{i-1}, \mathbf{S}^{i-1})$
$\mathbf{F}^{i}_{stat} \leftarrow \mathrm{ConvGRU}(\mathbf{F}^{i-1}_{stat}, \mathbf{F}^{i-1}_{mot}, \mathbf{F}_{ctx})$
$({}_\triangle\mathbf{D}, {}_\triangle\mathbf{G}, {}_\triangle\mathbf{O}, \boldsymbol{\delta}^i_U) \leftarrow \mathrm{Head}(\mathbf{F}^i_{stat})$
$\mathbf{D}^i_U \leftarrow \mathbf{D}^{i-1} +_\triangle \mathbf{D}$
$\mathbf{R}^i \leftarrow \Phi(\mathrm{cat}(\mathbf{D}^i_U, \mathbf{D}_{\mathrm{mono}}, \mathbf{F}_{\mathrm{ctx}}, \mathbf{F}^i_{\mathrm{stat}}))$
$\mathbf{S}^i \leftarrow \{\mathbf{G}^{i-1} +_\triangle \mathbf{G}, \mathbf{O}^{i-1} +_\triangle \mathbf{O}, \mathbf{R}^i\}$
$\mathbf{D}^i, \boldsymbol{\delta}^i \leftarrow \mathrm{LSGP}(\mathbf{D}^i_U, \boldsymbol{\delta}^i_U, \mathbf{S}^i, N_2)$
```

Motion Feature. The motion feature $\mathbf{F}^{i-1}_{mot}$ is generated by a motion feature encoder (MFE). In MFE, we encode the concatenated features $\mathrm{cat}(\mathbf{D}^{i-1}, \boldsymbol{\delta}^{i-1}, \mathbf{S}^{i-1})$ and a small dynamic cost volume $\mathbf{C}$ with two separate two-layer CNNs, then fuse them together with a single layer CNN, and concatenate the fused feature maps with $\mathrm{cat}(\mathbf{D}^{i-1}, \boldsymbol{\delta}^{i-1}, \mathbf{S}^{i-1})$ to obtain $\mathbf{F}^{i-1}_{mot}$. The small dynamic cost volume $\mathbf{C}$ with dimensions $G \times D \times H \times W$ is constructed by warping right image feature $\mathbf{F}_r$ and then computing group-wise correlations:

$$\mathbf{C}(g, d, h, w) = \frac{1}{C_g} \sum_{c=1}^{C_g} \mathbf{F}_l(c, h, w)\mathbf{F}_r(c, h', w'), \tag{7}$$

where $g \in [0, G-1]$ is the group index, $d \in [0, D-1]$ is the depth index, and $h' = h + f_h(d)$, $w' = w + f_w(d) - \mathbf{D}^{i-1}(h, w)$. We adopt a 2D-1D alternate local search strategy [18] where, in 1D search mode, $f_h(d) = 0$ and $f_w(d) \in [-4, 4]$, and in 2D search mode, $f_h(d) \in [-1, 1]$ and $f_w(d) \in [-1, 1]$. This setting results in $\mathbf{C}$ having a depth $D = 9$, and we empirically set $G = 8$.

Multi-level ConvGRU and LSGP. The multi-level ConvGRU operates on four levels simultaneously, namely 1/4, 1/8, 1/16 and 1/32 resolutions. The GRUs are based on separable convolutions [18, 37] and are cross-connected [20], implying that the hidden states of adjacent levels are also the inputs to the current level. As shown in Fig 3(d), LSGP works on resolution $(H^{\mathrm{p}}, W^{\mathrm{p}}) = (H, W)$ with N_2 iterations.

3.3.2 Upsampling and Refinement

We employ convex upsampling [37] to reconstruct the disparity map, uncertainty map and LSI to the original resolution, and then refine the disparity map with LSGP.

Upsampling. Convex upsampling treats the original resolution disparity values as a weighted sum of their coarse-resolution neighbors in a 3×3 grid. We apply an additional GRU updating to align hidden states $\mathbf{F}^N_{stat}$ and disparity map $\mathbf{D}^N$, resulting in $\mathbf{S}^{\mathrm{lr}}$, $\boldsymbol{\delta}^{\mathrm{lr}}$ and $\mathbf{D}^{\mathrm{lr}}$. Then, we upsample $\mathbf{S}^{\mathrm{lr}}$, $\boldsymbol{\delta}^{\mathrm{lr}}$ and $\mathbf{D}^{\mathrm{lr}}$ to obtain $\mathbf{S}^{\mathrm{hr}}$, $\boldsymbol{\delta}^{\mathrm{hr}}_U$ and $\mathbf{D}^{\mathrm{hr}}_U$.

Refinement with LSGP. we apply LSGP with N_2 iterations to obtain $\mathbf{D}^{\mathrm{hr}}$ according to $\mathbf{D}^{\mathrm{hr}}_U$, $\mathbf{S}^{\mathrm{hr}}$ and $\boldsymbol{\delta}^{\mathrm{hr}}_U$, as shown in Fig 3(e). Here, LSGP works on resolution $(H^{\mathrm{p}}, W^{\mathrm{p}}) = (sH, sW)$.

3.4. Supervision

Loss Functions. During training, we collect the intermediate outputs in disparity updating as a sequence, compute the $l1$ distance between the predicted values and the ground truth values. The total loss composes of three parts:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_d + \mathcal{L}_o + \mathcal{L}_g, \\ \mathcal{L}_d &= \textstyle\sum_{i=1}^{2N} \gamma^{2N-i} \|\mathbf{D}_{gt} - \mathbf{D}^i\|_1, \\ \mathcal{L}_o &= \textstyle\sum_{i=1}^{N} \gamma^{N-i} \|\mathbf{o}_{gt} - \mathbf{o}^i\|_1, \\ \mathcal{L}_g &= \textstyle\sum_{i=1}^{N} \gamma^{N-i} \|\mathbf{G}_{gt} - \mathbf{G}^i\|_1 \end{aligned} \tag{8}$$

where $\mathbf{D}^i \in \{\mathrm{CUp}(\mathbf{D}^1), \mathrm{CUp}(\mathbf{D}^1_U), \cdots, \mathbf{D}^{\mathrm{hr}}_U, \mathbf{D}^{\mathrm{hr}}\}$ and $\mathbf{o}^i$ is computed with Eq. 2 based on $\{\mathbf{O}^i, \mathbf{G}^i\} \subseteq \mathbf{S}^i \in \{\mathrm{CUp}(\mathbf{S}^1), \cdots, \mathbf{S}^{\mathrm{hr}}\}$. We set $\gamma = 0.9$ and CUp() stands for convex upsampling.

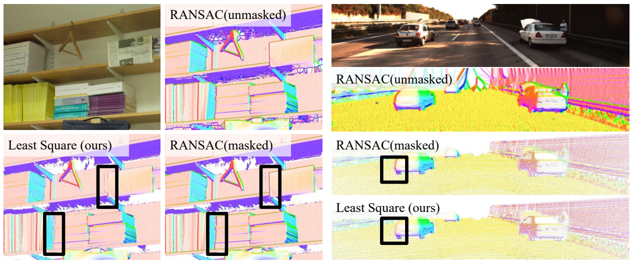


Figure 6. An illustration of the disparity gradient labels. RANSAC based method depends on densification pre-processing, and may introduce artifacts on edges (see black boxes). Our weighted least square solution directly works with semi-dense ground truth and there are less noises. Best zoomed in.

Labels Generation. The label $\mathbf{o}_{gt}$ is simply derived from $\mathbf{D}_{gt}$ with unfolding operator, while $\mathbf{G}_{gt}$ is generated by solving a weighted least square problem. For a pixel $\mathbf{p}$, we compute the disparity gradient within a 9×9 window $\mathcal{N}_9(\mathbf{p})$ centered at $\mathbf{p}$:

$$\mathbf{G}_{gt}(\mathbf{p}) = \arg\min_{\mathbf{G}(\mathbf{p})} \sum_{\mathbf{p}_j \in \mathcal{N}_9(\mathbf{p})} \mathbf{w}(\mathbf{p}, \mathbf{p}_j)(\mathbf{G}(\mathbf{p}) \cdot \Delta\mathbf{p}_j - \mathbf{o}_{gt}(\mathbf{p}, \mathbf{p}_j))^2, \tag{9}$$

where $\mathbf{w}$ is the weight:

$$\mathbf{w}(\mathbf{p}, \mathbf{p}_j) = \mathbf{M}(\mathbf{p}, \mathbf{p}_j)\exp(-\mathbf{d}(\mathbf{p}, \mathbf{p}_j)), \tag{10}$$

where $\mathbf{d}(\mathbf{p}, \mathbf{p}_j) = \|\mathbf{p} - \mathbf{p}_j\|_2^2 + (\mathbf{D}_{gt}(\mathbf{p}) - \mathbf{D}_{gt}(\mathbf{p}_j))^2$. $\mathbf{M}$ is a binary mask, which is set to 1 if and only if both $\mathbf{D}_{gt}(\mathbf{p})$ and $\mathbf{D}_{gt}(\mathbf{p}_i)$ are valid. Compared to the RANSAC-based implementation used in HITNet [36], our disparity gradient label generation method offers two advantages: 1) It is efficient since the least square problem has a closed-form solution, while the RANSAC-based method requires several iterations. 2) It can work with semi-dense ground truth directly, without the need for any densification pre-processing. The disparity gradient label examples are shown in Fig. 6.

4. Experiments

In this section, We mainly evaluate our LoS on different benchmarks and analyse the LSGP. The ablation studies are contained in the supplementary.

4.1. Benchmark Evaluations

We evaluate our LoS on four popular public benchmarks, the quantitative results are shown in Table 1 and visual comparison results are illustrated in Fig. 7.

Datasets. We evaluate the proposed LoS on four popular public benchmarks, including ETH3D [30], Middlebury [29], KITTI 2012 [11] and KITTI 2015 [25]. For model training, we collect data from various public datasets to compose the basic training set (BTS), including SceneFlow [24], CRE [18], MPI-Sintel [5], FallingThings [39] and Instereo2K [1]. There are $300,028$ samples in the BTS.

Implementation Details. We implement our model using Pytorch [27] and train the model with the AdamW optimizer [21]. The training process consists of $300k$ steps, and the input data is 640×480 with a batch size of 16. We set the max learning rate to $4e^{-4}$, the learning rate linearly warms up from 5% to 100% in the first $6k$ steps (2%). Then, after $180k$ steps (60%), the learning rate is linearly decreased from 100% to 5%. We set $N = 5$, $N_1 = 32$, $N_2 = 4$ during training and $N = 10$, $N_1 = 64$, $N_2 = 4$ for test.

ETH3D. Following CREStereo [18], we train the model from scratch without fine-tuning. The full training set is composed of the BTS and ETH3D training set, with ETH3D being augmented to 2% of the full training set. We achieve the best performance among all of the published methods in terms of Bad1.0 metrics and achieve the state-of-the-art performance on AvgErr metrics. For ETH3D, most of the challenging areas belong to classes 3 and 4, which are covisible for the stereo pairs. Thus, our LSGP achieves consistent improvement when being tested with all pixels and with only non-occluded pixels.

Middlebury. The training strategy on Middlebury is the same as ETH3D and the full training set is the combination of the BTS and Middlebury training set. LoS outperforms all existing methods in terms of AvgErr-all metric. Since our LoS updates the disparities in challenging areas for dozens of times with LSGP, it achieves the lowest average error. Compared to DLNR [51], our LoS achieves comparable performance but is more efficient. Specifically, LoS runs at 0.93 s/frame on RTX4090 while DLNR runs at 1.68 s/frame on Tesla A100.

KITTI 12 & KITTI 15. We first train the models with the BTS and then finetune the models for $50k$ steps with the combination of KITTI 12, KITTI 15 and BTS. The KITTIs account for 80% in the finetuning dataset and the rest are randomly sampled from BTS. During finetuning, the max learning rate is $1e^{-4}$. We rank the first when testing within reflective regions on KITTI 2012, because most of the reflective regions are from cars, which can be handled well by LSGP. Additionally, we also achieve state-of-the-art performance on other metrics of KITTI 12 & 15, which shows the superiority of our LoS.

Robust Vision Challenge. Following [16], we first train the model with ETH3D, Middlebury and BTS and then introduce KITTIs to finetune the model for $50k$ steps. During finetuning, we augment the KITTIs to 50% and set the max learning rate to $1e^{-4}$. We achieve the best overall performance within the robust vision challenge settings, which demonstrates that local structure guidance benefits the robustness of stereo matching.

4.2. Analyse LSGP

Efficiency Evaluation. We conduct an efficiency comparison among RAFT-Stereo, CREStereo, IGEV, and LoS under their default inference settings, as summarized in Ta-

Method	ETH3D				Middlebury				KITTI 2015				KITTI 2012			
	Bad1.0	AvgErr	Bad1.0	AvgErr	Bad2.0	AvgErr	Bad2.0	AvgErr	D1-bg	D1-fg	D1-all	D1-all	out-3	out-3	out-3	out-3
	all	all	noc	noc	all	all	noc	noc	all	all	all	noc	all	noc	all(R)	noc(R)
PSMNet [8]	-	-	-	-	-	-	-	-	1.86	4.62	2.32	2.14	1.89	1.49	10.18	8.36
GwcNet [12]	-	-	-	-	-	-	-	-	1.74	3.93	2.11	1.92	1.70	1.32	9.28	7.80
LEAStereo [9]	-	-	-	-	12.1	2.89	7.15	1.43	1.40	2.91	1.65	1.51	1.45	1.13	6.50	5.35
AdaStereo [34]	3.34	0.25	3.09	0.24	19.8	3.39	13.7	2.22	2.59	5.55	3.08	2.83	-	-	-	-
HITNet [36]	3.11	0.22	2.79	0.20	12.8	3.29	6.46	1.71	1.74	3.20	1.98	1.74	1.89	1.41	7.54	5.91
CFNet [31]	3.70	0.26	3.31	0.24	-	-	-	-	1.54	3.56	1.88	1.73	1.58	1.23	7.29	5.96
RAFT-Stereo [20]	2.60	0.19	2.44	0.18	9.37	2.71	4.74	1.27	1.58	3.05	1.82	1.69	1.66	1.30	6.48	5.40
PCWNet [32]	-	-	-	-	-	-	-	-	**1.37**	3.16	1.67	1.53	**1.37**	**1.04**	6.20	4.99
ACVNet [45]	2.86	0.24	2.58	0.23	19.5	12.1	13.7	2.22	**1.37**	3.07	1.65	1.52	1.47	1.13	8.67	7.03
CREStereo [18]	1.09	**0.14**	0.98	**0.13**	8.13	2.10	3.71	1.15	1.45	2.86	1.69	1.54	1.46	1.14	7.27	6.27
DLNR [51]	-	-	-	-	**6.98**	1.91	**3.20**	**1.06**	1.60	**2.59**	1.76	1.61	-	-	-	-
IGEV [46]	1.51	0.20	1.12	0.14	8.16	3.64	4.83	2.89	1.38	2.67	**1.59**	**1.49**	1.44	1.12	5.00	4.35
CroCo-Stereo [43]	1.14	0.15	0.99	0.14	11.1	2.36	7.29	1.76	1.38	2.65	**1.59**	1.51	-	-	-	-
LoS(Ours)	**1.03**	0.15	**0.91**	0.14	8.03	**1.75**	4.20	1.12	1.42	2.81	1.65	1.52	1.38	1.10	**4.45**	**3.47**
iResNet_ROB [19]	4.67	0.27	4.23	0.25	31.7	6.56	24.8	4.51	2.27	4.89	2.71	2.40	-	-	-	-
CFNet_RVC [31]	3.70	0.26	3.31	0.24	16.1	5.07	10.1	3.49	1.65	3.53	1.96	1.76	-	-	-	-
CREStereo++_RVC [16]	1.70	0.16	1.59	0.15	9.46	**2.20**	**4.68**	**1.28**	**1.55**	3.53	1.88	1.75	-	-	-	-
LoS_RVC(Ours)	**1.47**	**0.14**	**1.26**	**0.13**	**9.30**	2.36	5.14	1.57	1.58	**3.08**	**1.83**	**1.71**	-	-	-	-

Table 1. Results on four popular benchmarks. **Top**: Comparison with fine-tuned models. **Bottom**: Comparison with winners of the Robust Vision Challenges 2018, 2020 and 2022. The second and third rows show the metrics and testing masks. All metrics are presented in percentages except for AvgErr presented by pixels. For testing masks, "all" denotes being tested with all pixels while "noc" denotes being tested with a non-occlusion mask. "(R)" denotes being tested within reflective regions. The **best** and second best are marked with colors.

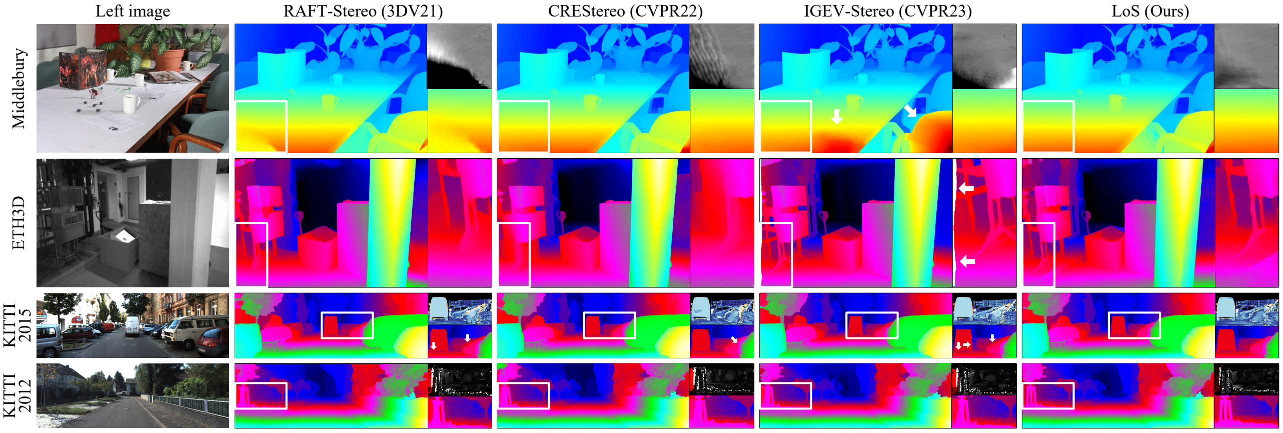


Figure 7. Qualitative results on the test set of Middlebury, ETH3D, KITTI2015, and KITTI2012. Since ETH3D benchmark do not provide the error map, we only show the zoomed highlight parts. Best zoomed in.

	RAFT-Stereo [20]	CREStereo [18]	IGEV [46]	LoS (Ours)
960 × 640	0.43s	0.41s	0.47s	0.31s
1920 × 1080	1.21s	0.99s	1.09s	0.73s

Table 2. Time consumption comparison between typical optimization based methods. All the models are tested with the officially released codes and default inference settings on a RTX4090 GPU.

ble 2. Our LoS is more efficient than the counterparts while achieves better cross-dataset overall performance. This efficiency improvement stems from the introduction of LSGP, which are more efficient (as depicted in Fig 5(b)), to reduce GRU iterations.

Challenging Areas Evaluation. We evaluate the RAFT-Stereo, IGEV and LoS on UnrealStereo4K [38] dataset with four category masks. The process of mask generation is detailed in supplementary. We use the parameters trained with Middlebury, and since CREStereo [18] do not release their Middlebury weights, we exclude CREStereo in this evaluation. As shown in Table 3, our LoS significantly outperforms the counterparts on AvgErr metric, and achieves comparable performance in terms of Bad2.0, which is consistent with the results in Table 1. Additionally, LoS significantly surpass the LoS model without LSGP in all metrics. The results demonstrate that LSGP markedly improves the dis-

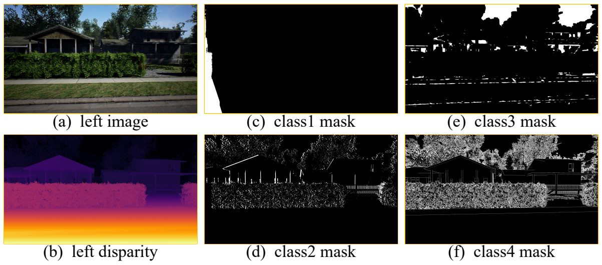

(a) left image
(c) class1 mask
(e) class3 mask
(b) left disparity
(d) class2 mask
(f) class4 mask

Figure 8. Test sample from UnrealStereo4K dataset.

	time (s)	overall	class1 (4.62%)	class2 (3.72%)	class3 (27.9%)	class4 (5.74%)
RAFT-Stereo	3.34	15.7 / 14.11	36.3 / 32.15	**47.5** / 16.54	**23.5** / 19.96	**48.2** / 20.81
IGEV	3.90	25.8 / 22.23	43.0 / 41.53	48.5 / 20.58	38.2 / 26.73	52.3 / 25.93
LoS	1.85	**15.0 / 6.28**	**29.1 / 9.05**	48.8 / **13.81**	26.6 / **7.08**	51.3 / **13.36**
(w/o LSGP)	1.44	19.9 / 18.81	55.0 / 18.90	67.5 / 49.59	31.6 / 26.51	55.7 / 27.19

Table 3. Evaluation with different category masks on UnrealStereo4K. We report Bad2.0(%)/AvgErr(pixel) metrics for each method and class. We also report the average ratio for each class.

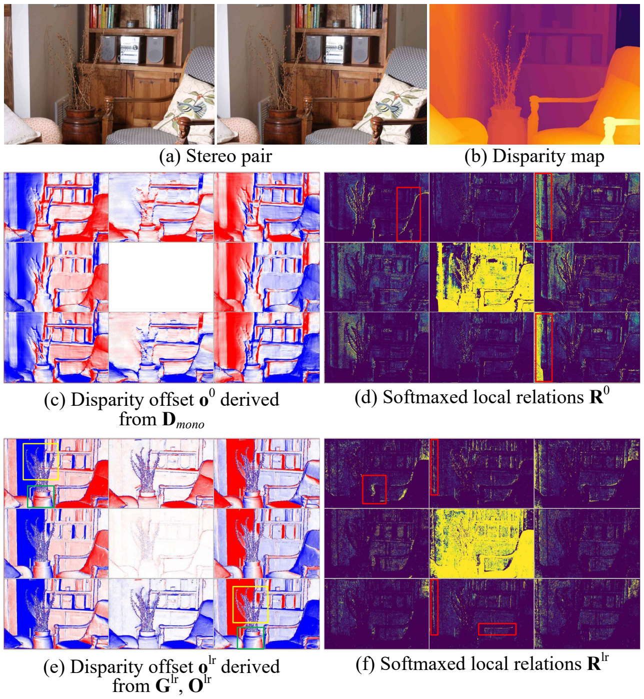

(a) Stereo pair
(b) Disparity map
(c) Disparity offset $\mathbf{o}^0$ derived from $\mathbf{D}_{mono}$
(d) Softmaxed local relations $\mathbf{R}^0$
(e) Disparity offset $\mathbf{o}^{\mathrm{lr}}$ derived from $\mathbf{G}^{\mathrm{lr}}$, $\mathbf{O}^{\mathrm{lr}}$
(f) Softmaxed local relations $\mathbf{R}^{\mathrm{lr}}$

Figure 9. An illustration of LSI. We visualize $\mathbf{o}$ and $\mathbf{R}$ of two stages. Since $\mathbf{o}$ and $\mathbf{R}$ are the representations for pixel-pairs $\{(\mathbf{p}, \mathbf{p}_i)|\mathbf{p}_i \in \mathcal{N}(\mathbf{p})\}$, we arrange the images into a 3×3 grid according to the relative position between $\mathbf{p}$ and $\mathbf{p}_i$. For clarity, we limit $|\mathbf{o}| \leq 0.1$ and $\mathrm{softmax}(\mathbf{R}) \leq 0.5$. Best zoomed in.

parity estimation accuracy in challenging areas, especailly the class 1 region.

Understanding LSI. We visualize the LSI in Fig. 9. The local relation constrains pixels to derive disparities from neighboring pixels belonging to the same object. For example, the third column of (d) and (f) are brighter on the left side, which means the pixels on the left side of the image (class1 regions) tend to obtain information from the right neighbors. On the contrary, the pixels on the left side of foreground objects (class2 regions) tend to obtain information from their left neighbors, resulting in the brighter regions in the first column of (d) and (f). It is also evident that the disparity offsets $\mathbf{o}$ effectively manage boundaries and curved surfaces, indicated by the green boxes in Fig. 9 (e). Comparing Fig. 9 (c) and (e), $\mathbf{o}$ optimized by GRU and is more accurate, which benefits the LSGP. The ablation study also highlights the substantial enhancement in LSI quality achieved through the iterative updating by GRU, consequently resulting in a notable improvement in performance. Please refer to the supplementary material for further details.

Limitations. First, the LSI struggles with extremely complex structures, such as these structures involving inter-occlusions, as indicated by the yellow boxes in Fig. 9 (e). These complex structures are also prevalent in KITTI, including dense vegetation and wooded areas, which limit the performance of LoS. Second, there are no constraints imposed on disparity offset residuals $\mathbf{O}$ currently, leading to residuals in self-disparity offset $\mathbf{o}(\mathbf{p}, \mathbf{p})$ after several GRU updating, as demonstrated in the central subplot of Fig. 9 (e). These residuals cause LSGP to partially replicate the role of GRU updating (step 4 of Alg. 1), which constrains our ability to adjust N_2 freely during inference. Third, despite our efforts to mitigate the impact of high-uncertainty pixels by splitting the neighbors (Eq. 3), these pixels may still affect pixels with low uncertainty, marginally diminishing our method's performance in terms of Badx.0 metric.

5. Conclusion

In this paper, we propose LoS, an optimization-based stereo matching method enhanced by local structure guidance. We first present the local structure of a scene with an extended LSI to capture more details and handle non-planar structures such as curve faces and object boundaries. Then we propose LSGP on the basis of LSI to update the estimated disparity map with local structure guidance. Despite the limitations, our LSI introduces informative local structure guidance to stereo matching, and LSGP significantly and efficiently improves the disparity accuracy under the structure guidance. Extensive experiments on four popular benchmarks and robust vision challenge demonstrate the effectiveness, robustness and efficiency of the proposed LoS.

Acknowledgement. This work was partially supported by the Shenzhen Science and Technology Program (No. RCYX20200714114641140), National Natural Science Foundation of China (No. U20A20185, 62372491, 62301601), and the Guangdong Basic and Applied Basic Research Foundation (2022B1515020103).

References

[1] Wei Bao, Wei Wang, Yuhua Xu, Yulan Guo, Siyu Hong, and Xiaohu Zhang. Instereo2k: a large real dataset for stereo matching in indoor scenes. *Science China Information Sciences*, 63:1–11, 2020. 6

[2] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 2

[3] Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3.1 – a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023. 4

[4] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 1–11, 2011. 1, 2

[5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 611–625. Springer-Verlag, 2012. 6

[6] Ayan Chakrabarti, Ying Xiong, Steven J. Gortler, and Todd Zickler. Low-level vision by consensus in a spatial hierarchy of regions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[7] Ayan Chakrabarti, Ying Xiong, Steven J. Gortler, and Todd Zickler. Low-level vision by consensus in a spatial hierarchy of regions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[8] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5418, 2018. 2, 7, 3

[9] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 22158–22169, 2020. 7

[10] Shivam Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4384–4393, 2019. 2

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 6, 3

[12] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3273–3282, 2019. 2, 7

[13] Yulan Guo, Yun Wang, Longguang Wang, Zi Wang, and Chen Cheng. Cvcnet: Learning cost volume compression for efficient stereo matching. *IEEE Transactions on Multimedia (TMM)*, 2022. 2

[14] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2360–2367, 2013. 1, 2

[15] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):328–341, 2007. 2

[16] Junpeng Jing, Jiankun Li, Pengfei Xiong, Jiangyu Liu, Shuaicheng Liu, Yichen Guo, Xin Deng, Mai Xu, Lai Jiang, and Leonid Sigal. Uncertainty guided adaptive warping for robust and efficient stereo matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 6, 7

[17] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 66–75, 2017. 2

[18] Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16263–16272, 2022. 1, 2, 5, 6, 7, 3

[19] Zhengfa Liang, Yulan Guo, Yiliu Feng, Wei Chen, Linbo Qiao, Li Zhou, Jianfeng Zhang, and Hengzhu Liu. Stereo matching using multi-level cost volume and multi-scale feature constancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(1):300–315, 2019. 2, 7

[20] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *Proceedings of the IEEE International Conference on 3D Vision (3DV)*, pages 218–227. IEEE, 2021. 1, 2, 4, 5, 7, 3

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[22] Yamin Mao, Zhihua Liu, Weiming Li, Yuchao Dai, Qiang Wang, Yun-Tae Kim, and Hong-Seok Lee. Uasnet: Uncertainty adaptive sampling network for deep stereo matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6311–6319, 2021. 2

[23] David Marr and Tomaso Poggio. Cooperative computation of stereo disparity: A cooperative algorithm is derived for extracting disparity information from stereo image pairs. *Science*, 194(4262):283–287, 1976. 2

[24] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016. 2, 6, 3

[25] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6, 3

[26] Guang-Yu Nie, Ming-Ming Cheng, Yun Liu, Zhengfa Liang, Deng-Ping Fan, Yue Liu, and Yongtian Wang. Multi-level

context ultra-aggregation for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3283–3291, 2019. 2

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. 6

[28] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(3), 2022. 4

[29] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*, pages 31–42. Springer, 2014. 6, 3

[30] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6, 3

[31] Zhelun Shen, Yuchao Dai, and Zhibo Rao. Cfnet: Cascade and fused cost volume for robust stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13906–13915, 2021. 2, 7, 3

[32] Zhelun Shen, Yuchao Dai, Xibin Song, Zhibo Rao, Dingfu Zhou, and Liangjun Zhang. Pcw-net: Pyramid combination and warping cost volume for stereo matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 280–297. Springer, 2022. 2, 7

[33] Sudipta N Sinha, Daniel Scharstein, and Richard Szeliski. Efficient high-resolution stereo matching using local plane sweeps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1582–1589, 2014. 2

[34] Xiao Song, Guorun Yang, Xinge Zhu, Hui Zhou, Zhe Wang, and Jianping Shi. Adastereo: A simple and efficient approach for adaptive stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10328–10337, 2021. 7

[35] Tatsunori Taniai, Yasuyuki Matsushita, Yoichi Sato, and Takeshi Naemura. Continuous 3d label stereo matching using local expansion moves. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(11):2725–2739, 2017. 1, 2

[36] Vladimir Tankovich, Christian Hane, Yinda Zhang, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14362–14372, 2021. 1, 2, 3, 6, 7

[37] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–419. Springer, 2020. 2, 5

[38] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8942–8952, 2021. 7, 1

[39] Jonathan Tremblay, Thang To, and Stan Birchfield. Falling things: A synthetic dataset for 3d object detection and pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2038–2041, 2018. 6

[40] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Itermvs: Iterative probability estimation for efficient multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8606–8615, 2022. 2

[41] Longguang Wang, Yulan Guo, Yingqian Wang, Zhengfa Liang, Zaiping Lin, Jungang Yang, and Wei An. Parallax attention for unsupervised stereo correspondence learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(4):2108–2125, 2020. 5

[42] Yun Wang, Longguang Wang, Hanyun Wang, and Yulan Guo. Spnet: Learning stereo matching with slanted plane aggregation. *IEEE Robotics and Automation Letters (RAL)*, 7(3):6258–6265, 2022. 1, 2, 3

[43] Philippe Weinzaepfel, Vaibhav Arora, Yohann Cabon, Thomas Lucas, Romain Brégier, Vincent Leroy, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Improved cross-view completion pre-training for stereo matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 7

[44] Bin Xu, Yuhua Xu, Xiaoli Yang, Wei Jia, and Yulan Guo. Bilateral grid learning for stereo matching networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12497–12506, 2021. 2

[45] Gangwei Xu, Junda Cheng, Peng Guo, and Xin Yang. Attention concatenation volume for accurate and efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12981–12990, 2022. 7

[46] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21919–21928, 2023. 1, 2, 7

[47] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1959–1968, 2020. 2

[48] Haofei Xu, Jiaolong Yang, Jianfei Cai, Juyong Zhang, and Xin Tong. High-resolution optical flow from 1d attention and correlation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10498–10507, 2021. 2

[49] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1592–1599, 2015. 2

[50] Feihu Zhang, Xiaojuan Qi, Ruigang Yang, Victor Prisacariu, Benjamin Wah, and Philip Torr. Domain-invariant stereo matching networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 420–439. Springer, 2020. 2, 3

[51] Haoliang Zhao, Huizhou Zhou, Yongjun Zhang, Jie Chen, Yitong Yang, and Yong Zhao. High-frequency stereo matching network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1327–1336, 2023. 1, 2, 6, 7