Beyond Row Alignment: Virtual-Camera-Aware Online Stereo Rectification

Yizhao Peng[1] **Rui Gong**[1] **Zaiwang Gu**[2] **Xudong Jiang**[1] **Jun Cheng**[2*]
[1] Nanyang Technological University, Singapore
[2] Institute of Advanced Intelligence and Computing, A*STAR, Singapore

Abstract

Stereo matchers estimate depth by comparing the left and right images from a stereo camera pair, but they assume that the two cameras are accurately calibrated and rectified. In real deployments, small physical shifts after calibration, caused by camera motion, mechanical drift, vibration, or installation changes, can break this assumption and degrade depth estimation. Online stereo rectification is therefore needed to correct the image pair during operation and recover a stereo geometry where standard stereo matching can work reliably. Recent online rectification methods mainly optimize row alignment, making the left and right views of the same scene point fall on the same image row. We show that this target, while necessary, is insufficient for deployable stereo depth estimation. A rectifier may make a small set of matched points look well row-aligned, but still distort the full images in ways that hurt stereo matching: useful image regions may be cut off, large blank areas may appear, objects may be stretched or squeezed, or image scale may change unevenly across the view. We propose VirtualRect, an online stereo rectification method designed for reliable real-world deployment. Rather than treating rectification as simply making the left and right image rows line up, this method estimates a rectified virtual stereo camera and combines image-wide supervision of image warp behavior. This makes the rectified images and the camera parameters come from the same geometry, so the images used for matching and the model used for depth computation remain consistent. Experiments on multiple datasets show that, compared with the prior state-of-the-art online rectifier, VirtualRect reduces vertical flow error by 17% on Carla-Flowguided and 7% on Semi-Truck Highway, and lowers depth MAE by 30–41% on Carla-Flowguided across three downstream stereo matchers.

1 Introduction

Online stereo rectification addresses a practical failure mode in stereo depth. Stereo matchers assume that the left and right images are well rectified: the same 3D scene point should appear on the same image row in both images, so matching can be reduced to horizontal disparity search [1, 2, 3]. This assumption usually comes from offline calibration. In real deployments, however, vibration, temperature changes, mechanical stress, or long-term structural deformation can shift the cameras away from their calibrated geometry. Once this happens, standard stereo matching becomes unreliable, and depth estimation degrades in systems such as mobile robots, smart glasses, and vehicle-mounted wide-baseline stereo systems [4, 5, 6, 7, 8, 9, 10]. Online rectification is therefore needed to recover usable rectified image pairs during operation.

Recent online rectifiers correctly put row alignment at the center of the problem. FloSR trains a real-time rectifier with vertical flow as supervision [9]. RsCeR replaces pose-first estimation with a rectification-constrained robust estimator and rectification-specific matcher supervision [10]. These

*Correspondence to: cheng_jun@a-star.edu.sg

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

works show that online rectification should optimize row alignment directly rather than treating rectification as a by-product of pose estimation. A rectifier may make a small set of matched points fall on the same rows, while still cutting off useful image regions, creating large blank areas, stretching objects, or changing image scale unevenly. In such cases, the row-alignment error can be small, but the rectified images can still be poor inputs for the downstream stereo matcher.

VirtualRect keeps the benefits of explicit matching and geometric solving, but makes the solver more aware of deployment needs. First, it supervises the predicted image warps by comparing their projections of an image-wide anchor grid with those produced by the pose-derived reference image warps, extending supervision beyond the observed matches. Second, after outlier rejection, it does not treat all remaining matches equally; instead, it reduces the influence of matches that are geometrically risky for the final rectification. Third, it deploys the result through a shared virtual camera, so that the rectified images and the depth computation remain consistent. As a result, VirtualRect estimates not just a row-aligning image transformation, but a rectified stereo camera that can be directly used by a downstream stereo-depth system.

Our contributions are:

- We introduce *metric-consistent virtual stereo deployment*, which treats the rectified virtual camera as part of the output and evaluates whether image formation and disparity-to-depth conversion are consistent.
- We introduce *image-wide homography-behavior supervision*, which supervises the predicted homographies by comparing their mappings of an image-wide anchor grid with those of pose-derived reference homographies, complementing pointwise correspondence residuals.
- We introduce *solver-aware observation calibration*, which attenuates geometrically risky post-consensus matches in the final constrained solve while keeping hypothesis generation driven by appearance confidence.
- Our method outperforms prior online rectification approaches under standard evaluation protocols and consistently improves performance across three different downstream stereo matchers.

2 Related Work

Pose estimation and robust two-view geometry. A common route to online stereo rectification is to estimate a relative pose and then derive rectifying homographies from that pose. Classical pipelines combine local features, minimal solvers, and robust estimators [11, 12, 13, 14]. Practical robust variants include LO-RANSAC, USAC, MAGSAC, and MAGSAC++ [15, 16, 17, 18]. Learned matchers improve the correspondence stage [19, 20, 21, 22]. Learned pose and scoring systems add neural components to geometry estimation [23, 24, 25, 26]. DirectionNet and 8-point ViT are representative learning-based pose baselines [27, 28]. These methods are powerful for camera geometry, but the objective is usually pose or epipolar consistency rather than the rectified image consumed by stereo matching. Our residual-risk module is closest in spirit to learned correspondence weighting, but it is used at a different point: after RANSAC selects a rectification-consensus set, it supplies an observation model for the final rectification solve rather than replacing the matcher or the minimal solver.

Stereo rectification and virtual-camera choice. Rectification is not unique: many homography pairs can make epipolar lines horizontal while producing different crops, distortions, and resampling patterns [1, 2, 3]. Classical work already recognized that legal rectifications should avoid harmful image formation: Loop and Zhang reduce distortion, Gluckman and Nayar minimize resampling effects, and Fusiello and Irsara restrict the solution to quasi-Euclidean transforms [29, 30, 31]. Subsequent analyses connect rectification choice to local discriminability and dense matching accuracy [32, 33, 34]. Online rectification brings this issue back in a deployed setting. FloSR uses vertical optical flow to train a real-time rectifier for wide-baseline stereo [9]. RsCeR directly optimizes rectification constraints with semi-dense matches and a constrained estimator [10]. VirtualRect asks the next deployment question: after rows are aligned, which rectified image formation and virtual camera should the downstream stereo system actually consume? Row alignment solves the search geometry, while virtual-camera deployment determines how disparity is interpreted as metric depth.

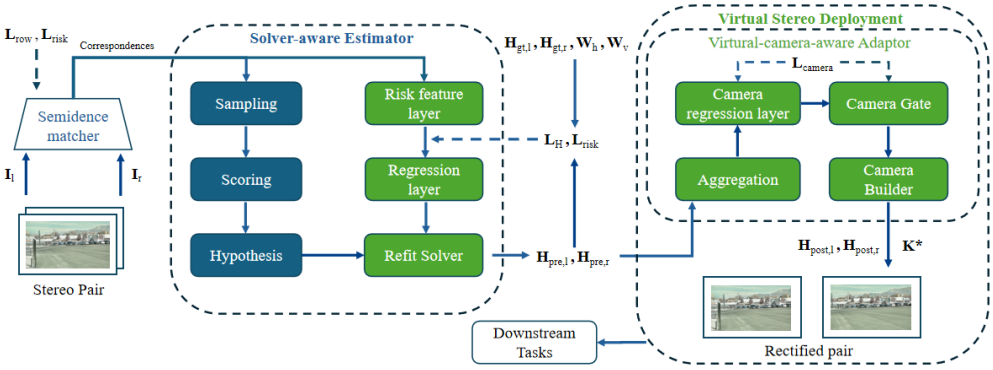

Figure 1: **VirtualRect Online Rectification.** The proposed framework consists of a semi-dense matcher, a solver-aware estimator, and a virtual stereo deployment module. The matcher takes the left and right images $\mathbf{I}_l, \mathbf{I}_r$ as input and outputs correspondences. The solver-aware estimator samples and scores for rectification hypotheses, predicts correspondence reliability, and solves for pre-deployment rectifying homographies $\mathbf{H}_l^{\text{pre}}, \mathbf{H}_r^{\text{pre}}$. The virtual-camera-aware adaptor then predicts a shared rectified camera $\mathbf{K}^*$ and deploys the final homographies $\mathbf{H}_l^*, \mathbf{H}_r^*$ with $\mathbf{K}^*$ for stereo matching.

Stereo matching and depth references. Modern stereo matchers such as DLNR, HITNet, and RAFT-Stereo assume a rectified stereo input and convert disparity into depth using a camera model [35, 36, 37]. Recent foundation stereo models, including Foundation Stereo, Monster, and Monster++, improve robustness and generalization for stereo depth [38, 39, 40]. Our method complements these stereo networks by selecting the rectified image formation and virtual camera used by a fixed matcher, ensuring that disparity-to-depth conversion uses the same camera model that generated the rectified images.

3 Method

Overview. VirtualRect takes a calibrated stereo pair as input and produces rectified images together with the virtual camera used for metric-depth recovery. During training, we supervise not only rectification error at matched correspondences but also how the predicted homographies map points across the image, by comparing their mappings of an image-wide anchor grid with those of pose-derived reference homographies. At inference, a semi-dense matcher first provides correspondences and appearance confidences. Next, a rectification-constrained solver estimates row-aligning homographies; RANSAC selects a consensus set, and the selected observations are calibrated using their predicted residual risks. Finally, a deployment head predicts a shared virtual camera and applies it to both views. The same virtual intrinsics are used to generate the rectified images and convert disparity to metric depth.

3.1 Image-Wide Homography Behavior Supervision

Pointwise row losses supervise where matches happen to be observed, but the deployed stereo matcher receives the full rectified image. A solver can fit sparse correspondences while producing a poor warp in low-texture road regions, image borders, thin structures, or other areas with weak match coverage. This mismatch is especially relevant in driving scenes, where useful depth often comes from texture-poor surfaces and object boundaries. We therefore supervise the emitted homographies as image transformations, not only the residuals of selected matches.

Let $\mathcal{A}$ be a regular grid of normalized anchor points covering the image and let $\pi(\cdot)$ denote homogeneous projection. The homography-behavior loss is

$$\mathcal{L}_H = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \left(\|\pi(\mathbf{H}_l a) - \pi(\mathbf{H}_l^{\text{gt}} a)\|_1 + \|\pi(\mathbf{H}_r a) - \pi(\mathbf{H}_r^{\text{gt}} a)\|_1 \right). \tag{1}$$

The loss applies the predicted and reference homographies to the same image-wide anchor grid and measures the distance between the transformed points. Unlike the sparse row loss, which is evaluated only at matched points, the anchor grid also evaluates the warp in regions without matches.

The supervision also changes what the network is rewarded for learning. A pointwise loss can be reduced by improving matches around visually easy regions; $\mathcal{L}_H$ is reduced only when the final solver emits a warp whose action agrees with the reference over the image plane. This is closer to the quantity used at deployment, because the downstream matcher never observes the latent pose or the inlier set. It observes the resampled images. We therefore treat the homography pair as an image-wide operator and supervise its behavior directly.

Another way to view Eq. (1) is as a sparse quadrature of the rectification map itself. The anchor grid does not introduce a second geometric target; it asks whether the same low-dimensional homography that fits the matches also produces a coherent image transformation over unsampled regions. This is why the term complements the constrained solver rather than weakening it: the solver still emits a physically structured rectification, while the loss makes the emitted operator visible to learning. In the ablation study, adding this term gives the largest single gain in vertical flow and pose consistency, which is why we treat it as the main geometric stabilizer of VirtualRect.

3.2 Solver-Aware Observation Calibration

For a correspondence $(\tilde{\mathbf{u}}_{l,i}, \tilde{\mathbf{u}}_{r,i})$, define its vertical rectification residual as

$$e_i(\mathbf{H}_l, \mathbf{H}_r) = y(\mathbf{H}_l \tilde{\mathbf{u}}_{l,i}) - y(\mathbf{H}_r \tilde{\mathbf{u}}_{r,i}), \tag{2}$$

where $y(\cdot)$ returns the inhomogeneous vertical coordinate after homogeneous normalization. The rectifying homographies depend on the solver parameters θ, so we also write this residual as $e_i(\theta)$. Robust consensus selects a plausible inlier set, but the final continuous solve still needs a reliability model for the selected observations. A uniform solver minimizes

$$\hat{\theta} = \arg\min_{\theta} \sum_{i \in \mathcal{I}} e_i(\theta)^2, \tag{3}$$

where θ parameterizes the five-degree-of-freedom constrained rectifier and $\mathcal{I}$ is the RANSAC inlier set. In wide-baseline driving, post-consensus observations are not homogeneous: repeated lane markings, occlusion boundaries, thin poles, rolling-shutter or synchronization artifacts, and support-boundary samples can pass a local consensus test while still biasing the continuous solve.

We therefore calibrate the observation precision used by the solver,

$$\hat{\theta} = \arg\min_{\theta} \sum_{i \in \mathcal{I}} w_i e_i(\theta)^2, \tag{4}$$

which yields the damped normal equation $(\mathbf{J}^\top \mathbf{W} \mathbf{J} + \lambda \mathbf{I})\delta = -\mathbf{J}^\top \mathbf{W} \mathbf{e}$. Here, $\mathbf{J} = \partial \mathbf{e} / \partial \theta$ is the residual Jacobian, $\mathbf{W} = \mathrm{diag}(w_i)$, and λ is the LM damping coefficient; the update is $\theta \leftarrow \theta + \delta$. Classical robust losses assign weights from residual magnitudes [41, 42, 43]. MAGSAC-style estimators marginalize over noise scales during hypothesis scoring [17, 18]. Our feature-conditioned calibration uses residual-risk weights to determine how strongly each selected observation affects the final continuous solve.

For each selected correspondence, a lightweight residual-risk head predicts a residual mean μ_i, Normal-Inverse-Gamma parameters $(\nu_i, \alpha_i, \beta_i)$, and a utility logit q_i. The supervised residual target is

$$r_i^{\mathrm{gt}} = y(\mathbf{H}_l^{\mathrm{gt}} \tilde{\mathbf{u}}_{l,i}) - y(\mathbf{H}_r^{\mathrm{gt}} \tilde{\mathbf{u}}_{r,i}), \tag{5}$$

with training loss

$$\mathcal{L}_{\mathrm{risk}} = \frac{1}{P} \sum_i \ell_{\mathrm{NIG}}(r_i^{\mathrm{gt}}; \mu_i, \nu_i, \alpha_i, \beta_i) + \lambda_e |r_i^{\mathrm{gt}} - \mu_i| (2\nu_i + \alpha_i). \tag{6}$$

At inference, the learned solver weights are clipped to $[0.05, 5]$. They do not replace RANSAC sampling or consensus selection and are applied only to the inliers selected by RANSAC during the final five-step LM refit.

3.3 Metric-Consistent Virtual Stereo Deployment

After the row-aligning solve and observation calibration, we obtain pre-deployment homographies $(\mathbf{H}_l^{\mathrm{pre}}, \mathbf{H}_r^{\mathrm{pre}})$ in solver intrinsic $\mathbf{K}$. A deployed stereo matcher, however, consumes rectified images and converts disparity to metric depth with a camera. If image formation and depth conversion use inconsistent virtual intrinsics, disparity and metric depth are inconsistent. Legal row-aligning rectifications can also expose different finite support and disparity ranges. The deployed output should therefore be a virtual stereo camera, not only a homography pair.

Classical rectification leaves freedom in the rectified image plane and intrinsic [2, 29, 30]. We restrict this freedom to a bounded shared gauge with two stable parameters: a focal-scale change η and a vertical principal-point shift Δc_y. They define

$$\mathbf{K}^* = \begin{bmatrix} e^{\eta} f & 0 & c_x \\ 0 & e^{\eta} f & c_y + \Delta c_y \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{G} = \mathbf{K}^* \mathbf{K}^{-1}. \tag{7}$$

The final deployed homographies are

$$\mathbf{H}_l^* = \mathbf{G}\mathbf{H}_l^{\mathrm{pre}}, \qquad \mathbf{H}_r^* = \mathbf{G}\mathbf{H}_r^{\mathrm{pre}}. \tag{8}$$

The same shared gauge is applied to both views, preserving the row alignment solution while choosing the output camera in which the rectified images are sampled. Metric depth uses the same $\mathbf{K}^*$ through $Z = f^* B/d$. This consistency is the reason $\mathbf{K}^*$ is part of the output rather than auxiliary metadata: if the images are generated in one virtual camera but disparity is interpreted in another, the stereo system is internally inconsistent. Deployment adjusts only the shared focal scale and vertical principal point, preserving the relative row-alignment solution while selecting a bounded virtual camera.

The deployment head takes a 19-D summary of solver and residual-risk statistics, pooled image features, and matching, weight, and coverage statistics. A $19 \to 128 \to 128 \to 2$ MLP with ReLU after each hidden layer predicts the camera proposal, and a separate $9 \to 1$ sigmoid gate predicts g. The proposal layer is initialized to zero. The bounded outputs are

$$\eta = 0.08 g \tanh(\tilde{\eta}), \qquad \Delta c_y = 16 g \tanh(\widetilde{\Delta c_y}), \qquad 0 < g < 1, \tag{9}$$

so $|\eta| \le 0.08$ and $|\Delta c_y| \le 16$ pixels. The output is restricted to the shared camera gauge in Eq. (7); it cannot change the relative row-alignment solution independently in the two views, and every change is recorded in the virtual intrinsic used for metric depth. The head is trained with three terms. The match term keeps selected correspondences row-compatible in the deployed camera,

$$\mathcal{L}_{\mathrm{match}} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \rho_\delta \left(y(\mathbf{H}_l^* \tilde{\mathbf{u}}_{l,i}) - y(\mathbf{H}_r^* \tilde{\mathbf{u}}_{r,i}) \right), \tag{10}$$

where ρ_δ is the Huber penalty. Let $S(\mathbf{H}_l, \mathbf{H}_r)$ be the common valid sampling support under the 5-pixel-border rule. The support and distortion penalties are

$$\mathcal{L}_{\mathrm{support}} = \max\{0, S(\mathbf{H}_l^{\mathrm{pre}}, \mathbf{H}_r^{\mathrm{pre}}) - S(\mathbf{H}_l^*, \mathbf{H}_r^*) - \epsilon_s\}, \quad \mathcal{L}_{\mathrm{dist}} = \eta^2 + \beta(\Delta c_y / H)^2. \tag{11}$$

The support term penalizes reductions in common valid area beyond ϵ_s, while $\mathcal{L}_{\mathrm{dist}}$ penalizes changes in focal scale η and vertical principal-point shift Δc_y.

3.4 Overall Loss Functions

Our overall loss function combines the above losses with the rectification-error loss.

Rectification Error Loss. We use the vertical rectification residual defined in Eq. (2). With known intrinsics, a calibrated rectifier can be written as

$$\mathbf{H}_l = \mathbf{K}\mathbf{R}_l\mathbf{K}_l^{-1}, \qquad \mathbf{H}_r = \mathbf{K}\mathbf{R}_r\mathbf{K}_r^{-1}, \tag{12}$$

where $\mathbf{K}_l$ and $\mathbf{K}_r$ are the original intrinsics, $\mathbf{K}$ is the shared target intrinsic, and $\mathbf{R}_l, \mathbf{R}_r$ rotate the two image planes into a common rectified plane.

At inference time, the constrained estimator minimizes an aggregate of Eq. (2) over the predicted correspondences within a tied-baseline five-degree-of-freedom rectifier family [5, 10]. During

training, we supervise the matcher using the ground-truth pose-induced rectifying homographies $(\mathbf{H}_l^{\mathrm{gt}}, \mathbf{H}_r^{\mathrm{gt}})$. The rectification-error loss is defined as

$$\mathcal{L}_{\mathrm{err}} = \frac{1}{P} \sum_{i=1}^{P} \left| \frac{\mathbf{H}_{l[2]}^{\mathrm{gt}} \tilde{\mathbf{u}}_{l,i}}{\mathbf{H}_{l[3]}^{\mathrm{gt}} \tilde{\mathbf{u}}_{l,i}} - \frac{\mathbf{H}_{r[2]}^{\mathrm{gt}} \tilde{\mathbf{u}}_{r,i}}{\mathbf{H}_{r[3]}^{\mathrm{gt}} \tilde{\mathbf{u}}_{r,i}} \right|, \tag{13}$$

where $\mathbf{H}_{[2]}$ and $\mathbf{H}_{[3]}$ denote the second and third rows of $\mathbf{H}$, respectively.

The pointwise loss $\mathcal{L}_{\mathrm{err}}$ measures vertical residuals at matched points, whereas $\mathcal{L}_H$ compares the predicted and reference homographies over the anchor grid.

This loss encourages the matcher to produce correspondences that are compatible with the rectification constraint optimized by the solver. Since the rectification-error loss mainly measures vertical consistency after rectification, we additionally report pose diagnostics: rotation and translation-direction errors indicate whether the recovered rectifying geometry remains geometrically coherent before virtual-camera deployment.

Final Loss. The two training stages expose the three method components:

$$\mathcal{L}_{\mathrm{solver}} = \mathcal{L}_{\mathrm{err}} + \mathcal{L}_{\mathrm{risk}} + \lambda_H \mathcal{L}_H, \qquad \mathcal{L}_{\mathrm{camera}} = \mathcal{L}_{\mathrm{match}} + \lambda_s \mathcal{L}_{\mathrm{support}} + \lambda_d \mathcal{L}_{\mathrm{dist}}. \tag{14}$$

At inference, the procedure remains explicit: match, robustly estimate a row-aligning solution, calibrate the final observation weights, and deploy the result as $(\mathbf{H}_l^*, \mathbf{H}_r^*, \mathbf{K}^*)$. The output is useful for downstream stereo provided that row alignment is maintained, sufficient image support is preserved, and the same virtual camera is used for image formation and metric-depth conversion. The three components therefore address different parts of the deployed rectification problem: operator-level supervision stabilizes the warp, observation calibration improves the final solve, and virtual-camera deployment makes the rectified images and metric depth belong to the same camera. Each component of the design acts on a different variable of the final deployed stereo object–the homography behavior, the observation precision, and the virtual camera.

4 Experiments

4.1 Datasets, Protocols, and Metrics

We use three benchmarks for the main evaluation. Carla-Flowguided and Semi-Truck Highway are the standard online rectification datasets introduced by FloSR and used by current SOTA RsCeR [9, 10]. Carla is synthetic with pose annotations, while Semi-Truck is a real wide-baseline driving dataset captured under vibration and structural stress. We also report pose diagnostics on Perturbed SceneFlow, a synthetic SceneFlow-derived benchmark with known perturbations and dense correspondence supervision, motivated by the same need to reduce dependence on pretrained evaluation metrics [10, 44]. Images from Carla and Semi-Truck are resized to 1024×512; Perturbed SceneFlow uses the SceneFlow evaluation resolution.

For the main benchmark results, we train a separate model on each benchmark's training split and evaluate it on the corresponding test split. The official splits do not share complete stereo pairs between training and testing, but do not require scenes or camera rigs to be disjoint.

Since Carla and Semi-Truck do not provide dense ground-truth correspondences, vertical flow remains the main dense rectification metric. And We employ a pretrained RAFT optical-flow model to estimate optical flow between stereo images, with the average magnitude of the vertical component used as the metric [45]. We also report sparse keypoint checks using SIFT and SuperGlue offsets. For the standard downstream-depth check, we follow the protocol of FloSR and RsCeR by using DLNR-Middlebury as the stereo matcher [9, 10, 35];

We also compute disparity coverage. We distinguish geometric support from downstream coverage. Let $\Omega = \{1, \ldots, W\} \times \{1, \ldots, H\}$ denote the full rectified image grid, and let $\Omega_{lr} \subset \Omega$ be the output pixels whose inverse samples are valid in both original images under the 5-pixel-border rule. All compared rectified pairs use the same left-to-right disparity convention. For a downstream matcher m with predicted disparity $d_m(p)$, we define disparity coverage as

$$\mathrm{Cov}_m = \frac{1}{HW} \sum_{p \in \Omega} \mathbf{1}[p \in \Omega_{lr}]\, \mathbf{1}[d_m(p) > 0]. \tag{15}$$

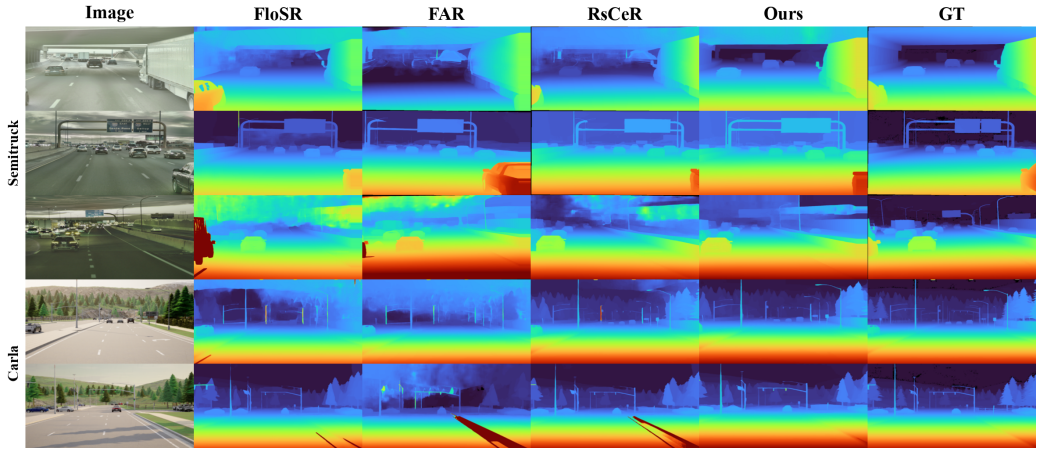

Figure 2: Qualitative downstream stereo disparity comparison by different online rectification methods on Semi-Truck Highway and Carla-Flowguided. Colored panels are disparity maps; the rightmost reference uses offline-calibration rectification.

It measures the fraction of the full rectified image on which the downstream matcher actually outputs a positive disparity. The datasets do not provide dense visibility annotations, so we do not apply a separate occlusion mask; all methods use the same border, disparity-positivity, reference-validity, and depth-threshold rules. Post-gauge depth is evaluated with the same virtual intrinsic that produced the post-gauge images.

4.2 Implementation Details

We implement VirtualRect in PyTorch with Kornia geometric operators [46, 47]. For controlled comparison, the matcher, constrained solver, data protocol, and evaluation configuration are held fixed to the RsCeR setting except for the learned observation calibration and virtual-camera deployment. The correspondence backbone is Efficient LoFTR pretrained on MegaDepth, with two matching layers, which offers a good accuracy–runtime trade-off for online rectification [22, 48]. Models are trained for 16,000 steps with batch size 16 on a single NVIDIA A40 GPU. The learning rate is 10^{-4} on Carla-Flowguided and 10^{-5} on Perturbed SceneFlow and Semi-Truck Highway, with linear annealing and an 800-step warm-up. During training and testing, RANSAC runs for 16 iterations with inlier threshold $\tau = 0.75$, and the Levenberg–Marquardt refit runs for 5 iterations. On Semi-Truck Highway, the filtering threshold is 0.3. Perturbed SceneFlow images are resized to 800×416. Optical-flow and keypoint-based evaluation follow the same configuration as prior online rectification work [9, 10].

The residual-risk head input concatenates a 64-D local descriptor from each image, the matcher confidence, and the 2-D fine displacement, for 131 dimensions in total. The head uses a $131 \to 128 \to 128 \to 5$ MLP with ReLU after each hidden layer. Softplus-based output transforms enforce $\nu_i > 0$, $\alpha_i > 1$, and $\beta_i > 0$.

Training is performed in two stages. In Stage 1, we train the residual-risk module with pointwise residual-scale supervision and image-wide homography-behavior supervision. Empirically, the latter is especially important because it constrains the learned solver by the induced warp itself, while the residual-risk term refines the observation model inside that stabilized solver. In Stage 2, we initialize the deployment head at identity and train the shared gauge correction that predicts the rectified virtual camera $\mathbf{K}^*$. Runtime is measured for the online rectifier at batch size 1 on one NVIDIA A40 and excludes downstream stereo inference.

4.3 Stereo Rectification Results

The final model is trained in two conceptual steps: residual-risk observation calibration and then virtual-camera deployment learning. Unless otherwise stated, the reported VirtualRect entries in the main tables are averaged over five random seeds. We organize the quantitative comparisons in the same reporting layout as RsCeR: rectification accuracy, downstream stereo checks, and pose

Table 1: **Stereo rectification accuracy.** We follow the keypoint-offset and vertical flow evaluation convention on Semi-Truck Highway and Carla-Flowguided. SIFT and SuperGlue offsets are independent sparse matcher checks. The VirtualRect row is the mean over five seeds. Arrows indicate that lower values are better.

Dataset	Method	SIFT Offset ↓ (pixels)	SuperGlue Offset ↓ (pixels)	Vertical Flow ↓ (pixels)
Semi-Truck Highway	SIFT + MAGSAC (2017) [11, 18]	21.10	1.15	1.37
	SP+SG + MAGSAC (2019) [18, 19, 20]	14.97	1.20	1.51
	LoFTR + MAGSAC (2021) [18, 21]	14.94	0.76	0.86
	EfficientLoFTR + MAGSAC (2024) [18, 22]	8.07	0.73	0.82
	DirectionNet (2021) [27]	8.41	6.10	6.44
	ViTPose / 8-point ViT (2022) [28]	4.53	2.37	2.83
	FAR (2024) [49]	4.20	0.49	0.75
	FloSR w/ OF (2024) [9]	4.01	0.59	1.05
	RsCeR (2025) [10]	3.65	0.43	0.59
	VirtualRect (Ours)	**3.10**	**0.42**	**0.55**
Carla-Flowguided	SIFT + MAGSAC (2017) [11, 18]	3.60	0.56	0.59
	SP+SG + MAGSAC (2019) [18, 19, 20]	5.77	1.15	1.36
	LoFTR + MAGSAC (2021) [18, 21]	7.29	0.57	0.64
	EfficientLoFTR + MAGSAC (2024) [18, 22]	3.12	0.59	0.62
	DirectionNet (2021) [27]	13.47	10.64	11.12
	ViTPose / 8-point ViT (2022) [28]	4.34	1.45	1.72
	FAR (2024) [49]	3.14	0.56	0.57
	FloSR w/ OF (2024) [9]	3.55	0.64	1.00
	RsCeR (2025) [10]	3.06	0.42	0.52
	VirtualRect (Ours)	**2.33**	**0.40**	**0.43**

Table 2: **Downstream stereo checks and disparity coverage on Carla-Flowguided.** Each rectified input is evaluated by three stereo matchers. For each matcher, Cov is Eq. (15). MAE is in meters, AbsRel and Cov are percentages, and Time reports the rectifier runtime only in milliseconds per stereo pair.

Method	Monster++			DLNR			HITNet			Time
	MAE ↓	AbsRel ↓	Cov ↑	MAE ↓	AbsRel ↓	Cov ↑	MAE ↓	AbsRel ↓	Cov ↑	
SIFT + MAGSAC [11, 18]	2.03	5.12	87.05	2.52	6.49	88.24	2.67	9.45	88.21	367
SP+SG + MAGSAC [18, 19, 20]	8.58	30.22	71.67	9.89	31.14	72.69	10.57	38.04	72.76	98
LoFTR + MAGSAC [18, 21]	3.60	9.90	86.25	4.13	11.10	87.45	4.22	14.31	87.41	237
EfficientLoFTR + MAGSAC [18, 22]	4.86	13.65	84.55	5.50	14.76	85.76	5.61	18.21	85.75	163
FAR [49]	3.85	9.94	86.71	4.35	11.03	87.94	4.37	13.81	87.61	1672
FloSR w/ OF [9]	5.72	20.15	81.39	6.16	20.23	82.56	6.43	25.01	82.52	81
RsCeR [10]	1.66	3.87	91.24	2.24	5.38	92.46	2.19	7.25	92.43	79
VirtualRect (Ours)	**0.98**	**2.11**	**97.47**	**1.50**	**3.31**	**98.52**	**1.54**	**5.13**	**98.50**	106

diagnostics. The main measurement remains row-alignment-centric: VirtualRect improves the vertical flow metric used by online rectification protocols, and the aligned images remain useful for downstream stereo.

Table 1 reports the core rectification metrics. Relative to the previous rectification-constrained method, VirtualRect reduces vertical flow by 17% on Carla-Flowguided and by 7% on Semi-Truck Highway. The SIFT-offset improvement is an independent check because it does not reuse the dense flow model, DLNR, or the semi-dense matcher that supplies our correspondences.

Table 2 shows that the rectification improvement transfers across downstream stereo matchers rather than one evaluator on Carla-Flowguided. Relative to RsCeR, VirtualRect reduces MAE by 41%, 33%, and 30% for Monster++, DLNR, and HITNet, respectively. The coverage columns show the complementary effect: before metric-depth conversion, the deployed rectified images let each matcher output positive disparity over substantially more of the frame. Coverage is measured before metric-depth conversion, while the K^* ablation below tests whether the image warp and the camera used for depth conversion are deployed consistently.

Table 3 shows that the pose diagnostics support the row-alignment half of the argument: before virtual-camera deployment, the estimated geometry is already closer to the reference geometry, reducing RE by 74% and 24% on Carla-Flowguided and Perturbed SceneFlow, respectively.

Zero-shot domain shift and baseline comparison. We additionally evaluate Carla-trained models on Semi-Truck Highway without fine-tuning as Table 4 shows. This protocol differs from the

Table 3: **Pose diagnostics.** These report how well the pre-gauge geometry recovers the row-alignment pose. RE is the overall rotation angular error, R_x, R_y, and R_z are absolute axis-angle components, and TE is the translation-direction angular error; all are in degrees. Lower is better.

Method	Carla-Flowguided					Perturbed SceneFlow				
	RE ↓	R_x ↓	R_y ↓	R_z ↓	TE ↓	RE ↓	R_x ↓	R_y ↓	R_z ↓	TE ↓
EfficientLoFTR + MAGSAC [18, 22]	0.208	0.007	0.205	0.018	2.736	0.263	0.014	0.254	0.035	1.385
FAR [49]	0.157	0.007	0.153	0.017	2.727	0.203	0.011	0.199	0.021	1.290
RsCeR [10]	0.081	0.003	0.077	0.006	0.860	0.152	0.008	0.147	0.018	0.678
VirtualRect (Ours)	**0.021**	**0.001**	**0.016**	**0.001**	**0.258**	**0.115**	**0.005**	**0.111**	**0.014**	**0.460**

Table 4: **Carla-trained, zero-shot on Semi-Truck.** Models are trained on Carla-Flowguided and evaluated on Semi-Truck Highway without fine-tuning. SIFT and SuperGlue report keypoint offsets, and RAFT reports vertical flow. All errors are in pixels; lower is better.

Method	SIFT Offset ↓	SuperGlue Offset ↓	RAFT Vertical Flow ↓
FloSR [9]	14.00	5.24	10.65
RsCeR [10]	3.76	0.43	0.62
VirtualRect (Ours)	**3.10**	**0.42**	**0.57**

Table 5: **Zero-shot on Carla-Drift, a temporally continuous dataset.** Results on all 155 sequences (155,000 stereo pairs), with fixed stereo translation. Lower is better.

Method	Temporal state	VOF-I (px) ↓	DC-I (m) ↓
TESO [50]	Yes	0.07	0.51
VirtualRect (Ours)	No	**0.02**	**0.25**

benchmark-specific training used for Table 1. We also evaluate TESO [50], which introduced CARLA-Drift specifically for continuous online stereo-calibration evaluation. Existing rectification benchmarks mainly provide isolated image pairs, and no real dataset provides known time-varying decalibration together with frame-wise calibration and depth ground truth. To our knowledge, CARLA-Drift is currently the only public benchmark we found that combines long continuous stereo sequences, controlled accumulated extrinsic drift, frame-wise camera ground truth, and depth ground truth. It contains 155 sequences of 1,000 consecutive frames. The right-camera rotation follows a random walk of up to 0.01 degree per frame on each axis.

TESO maintains temporal calibration state, whereas VirtualRect processes each stereo pair independently, without fine-tuning or temporal state. Both methods use the same frames, undrifted references, and evaluation code. Following the benchmark definition, each *-I* metric is the per-frame method error minus the corresponding undrifted-reference error; lower is better. Table 5 reports VOF-I in pixels and DC-I in meters.

4.4 Ablation Study

We use two ablations to evaluate two design choices in VirtualRect: the estimator and the virtual-camera deployment.

Table 6 isolates Stage-1 supervision while keeping the final $\mathbf{K}^*$ deployment fixed. The checkmark layout separates the row objective, image-wide homography-behavior supervision, and risk calibration. $\mathcal{L}_H$ gives the largest single improvement in both vertical flow and pose consistency, confirming that solver behavior must be constrained at the homography level. Adding $\mathcal{L}_{\text{risk}}$ on top of this stabilized solver yields the best deployed rectification and pose metrics.

Table 7 isolates the deployment camera. Moving from pre-gauge to post-gauge changes the rectified image formation; interpreting the post-gauge images with the old camera makes metric depth worse. Using the corresponding $\mathbf{K}^*$ restores consistency and improves the downstream metric. Thus online rectification should not stop at row alignment: the deployed stereo system also needs the virtual camera that generated its rectified images.

The online rectifier spends 70 ms in matching, 31 ms in RANSAC and refit, and 5 ms in the gauge and K^* deployment head on one NVIDIA A40, giving 106 ms per stereo pair in total.

Table 6: **Stage-1 supervision ablation on Carla-Flowguided under final deployment.** All variants use the same gauge/$\mathbf{K}^*$ deployment. RE and TE are angular errors in degrees. Lower is better.

Supervision	$\mathcal{L}_{\mathrm{row}}$	$\mathcal{L}_H$	$\mathcal{L}_{\mathrm{risk}}$	Vertical Flow ↓	RE ↓	TE ↓
Row alignment	✓	–	–	0.61	0.036	0.432
+ behavior supervision	✓	✓	–	0.48	0.025	0.288
+ inlier calibration	✓	✓	✓	**0.43**	**0.021**	**0.258**

Table 7: **Virtual-camera deployment ablation on Carla-Flowguided with DLNR.** Pre-gauge is the solver output; post-gauge uses the deployed virtual-camera warp. MAE is in meters and AbsRel is in percent. Lower is better.

RsCeR			VirtualRect		
Deployment	MAE ↓	AbsRel (%) ↓	Deployment	MAE ↓	AbsRel (%) ↓
pre-gauge + old $\mathbf{K}$	2.23	5.38	pre-gauge + old $\mathbf{K}$	1.61	3.86
post-gauge + old $\mathbf{K}$	3.21	9.25	post-gauge + old $\mathbf{K}$	3.00	9.03
post-gauge + $\mathbf{K}^*$	**2.14**	**4.86**	post-gauge + $\mathbf{K}^*$	**1.50**	**3.31**

5 Discussion and Limitation

Firstly, VirtualRect assumes known input intrinsics and restricts deployment to a bounded, shared focal-scale and vertical principal-point adjustment. Current online rectification benchmarks lack dense ground-truth rectified correspondences for this deployment setting, so image-wide warp behavior is supervised using pose-derived reference homographies and downstream performance is assessed indirectly.

But to date, public real-world datasets for online stereo rectification remain scarce, although they are needed to evaluate calibration drift under real deployment conditions. While Semi-Truck provides real-world data, we do not consider its measurement precision sufficient for a definitive assessment of this problem. Additional real-world sequences with accurately measured, time-varying stereo calibration and depth ground truth are needed to establish and test real-world long-term operation or SLAM stability.

Another interesting experiment in which we select the oracle virtual-camera position separately for each stereo pair yields substantially lower measured error than the current learned predictions. This suggests that the learned virtual-camera head might still have room for improvement.

6 Conclusion

We presented VirtualRect, a deployment-consistent online stereo rectification framework that estimates a rectified virtual stereo camera rather than only a pair of row-aligning homographies. Our key observation is that reliable stereo depth requires not only row alignment, but also valid rectified image formation and a camera model that is consistent with disparity-to-depth conversion. To this end, VirtualRect combines image-wide homography-behavior supervision, solver-aware observation calibration, and metric-consistent virtual-camera deployment. These components stabilize the rectification solver, reduce the influence of geometrically risky matches, and ensure that the rectified images and metric depth computation share the same geometry. Experiments on standard online rectification benchmarks show that VirtualRect improves rectification accuracy, downstream stereo depth, and disparity coverage across multiple stereo matchers. These results suggest that online rectification should be treated as a full stereo-camera deployment problem, not merely as row alignment.

Acknowledgments and Disclosure of Funding

This work was supported in part by the Agency for Science, Technology and Research (A*STAR) under its MTC Programmatic Funds (Grant No. M23L7b0021) and the National Health Innovation Centre Grant (NHIC-I2D-2502365).

References

[1] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

[2] Richard I. Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision*, 35(2):115–127, 1999.

[3] Andrea Fusiello, Emanuele Trucco, and Alessandro Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12:16–22, 2000.

[4] Thao Dang, Christian Hoffmann, and Christoph Stiller. Continuous stereo self-calibration by camera parameter tracking. *IEEE Transactions on Image Processing*, 18(7):1536–1550, 2009.

[5] Peter Hansen, Hatem Alismail, Peter Rander, and Brett Browning. Online continuous stereo extrinsic parameter estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1059–1066, 2012.

[6] Yonggen Ling and Shaojie Shen. High-precision online markerless stereo extrinsic calibration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1771–1778, 2016.

[7] Eike Rehder, Christian Kinzig, Philipp Bender, and Martin Lauer. Online stereo camera calibration from scratch. In *IEEE Intelligent Vehicles Symposium*, pages 1694–1699, 2017.

[8] Jialiang Wang, Daniel Scharstein, Akash Bapat, Kevin Blackburn-Matzen, Matthew Yu, Jonathan Lehman, Suhib Alsisan, Yanghan Wang, Sam Tsai, Jan-Michael Frahm, et al. A practical stereo depth system for smart glasses. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21498–21507, 2023.

[9] Anush Kumar, Fahim Mannan, Omid Hosseini Jafari, Shile Li, and Felix Heide. Flow-guided online stereo rectification for wide baseline stereo. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15375–15385, 2024.

[10] Rui Gong, Kim-Hui Yap, Weide Liu, Xulei Yang, and Jun Cheng. Rectification-specific supervision and constrained estimator for online stereo rectification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22348–22358, 2025.

[11] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[12] Richard I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

[13] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–777, 2004.

[14] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[15] Ondrej Chum, Jiri Matas, and Josef Kittler. Locally optimized ransac. In *DAGM Symposium*, pages 236–243, 2003.

[16] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. USAC: A universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):2022–2038, 2013.

[17] Daniel Barath, Jiri Matas, and Jana Noskova. MAGSAC: Marginalizing sample consensus. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10197–10205, 2019.

[18] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1304–1312, 2020.

[19] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 337–33712, 2018.

[20] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020.

[21] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8922–8931, 2021.

[22] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient loftr: Semi-dense local feature matching with sparse-like speed. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21666–21675, 2024.

[23] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. ASpanFormer: Detector-free image matching with adaptive span transformer. In *European Conference on Computer Vision*, pages 20–36, 2022.

[24] Johan Edstedt, Qiyu Sun, Georg Bokman, Marten Wadenback, and Michael Felsberg. RoMa: Robust dense feature matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024.

[25] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *IEEE/CVF International Conference on Computer Vision*, pages 4322–4331, 2019.

[26] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tian Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *IEEE/CVF International Conference on Computer Vision*, pages 5845–5854, 2019.

[27] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-baseline relative camera pose estimation with directional learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3258–3268, 2021.

[28] Chris Rockwell, Justin Johnson, and David F. Fouhey. The 8-point algorithm as an inductive bias for relative pose prediction by ViTs. In *International Conference on 3D Vision*, pages 1–11, 2022.

[29] Charles T. Loop and Zhengyou Zhang. Computing rectifying homographies for stereo vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 125–131, 1999.

[30] Joshua Gluckman and Shree K. Nayar. Rectifying transformations that minimize resampling effects. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages I–111–I–117, 2001.

[31] Andrea Fusiello and Luca Irsara. Quasi-euclidean uncalibrated epipolar rectification. In *International Conference on Pattern Recognition*, pages 1–4, 2008.

[32] John Mallon and Paul F. Whelan. Projective rectification from the fundamental matrix. *Image and Vision Computing*, 23(7):643–650, 2005.

[33] Martin Matousek, Radim Sara, and Vaclav Hlavac. Data-optimal rectification for fast and accurate stereovision. In *Third International Conference on Image and Graphics*, pages 212–215, 2004.

[34] Jianchen Liu, Bingxuan Guo, Wanshou Jiang, Weishu Gong, and Xiongwu Xiao. Epipolar rectification with minimum perspective distortion for oblique images. *Sensors*, 16(11):1870, 2016.

[35] Haoliang Zhao, Huizhou Zhou, Yongjun Zhang, Jie Chen, Yitong Yang, and Yong Zhao. High-frequency stereo matching network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1327–1336, 2023.

[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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14362–14372, 2021.

[37] Lahav Lipson, Zachary Teed, and Jia Deng. RAFT-Stereo: Multilevel recurrent field transforms for stereo matching. In *International Conference on 3D Vision*, pages 218–227, 2021.

[38] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundation-stereo: Zero-shot stereo matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5249–5260, 2025.

[39] Junda Cheng, Longliang Liu, Gangwei Xu, Xianqi Wang, Zhaoxing Zhang, Yong Deng, Jinliang Zang, Yurui Chen, Zhipeng Cai, and Xin Yang. Monster: Marry monodepth to stereo unleashes power. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6273–6282, 2025.

[40] Junda Cheng, Wenjing Liao, Zhipeng Cai, Longliang Liu, Gangwei Xu, Xianqi Wang, Yuzhou Wang, Zikang Yuan, Yong Deng, Jinliang Zang, Yangyang Shi, Jinhui Tang, and Xin Yang. Monster++: Unified stereo matching, multi-view stereo, and real-time stereo with monodepth priors. *arXiv preprint arXiv:2501.08643*, 2025.

[41] Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1): 73–101, 1964.

[42] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[43] Jonathan T. Barron. A general and adaptive robust loss function. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019.

[44] Nikolaus Mayer, Eddy Ilg, Philipp Haeusser, 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[45] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419, 2020.

[46] 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. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037, 2019.

[47] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: An open source differentiable computer vision library for pytorch. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3674–3683, 2020.

[48] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[49] Chris Rockwell, Nilesh Kulkarni, Linyi Jin, Jeong Joon Park, Justin Johnson, and David F. Fouhey. FAR: Flexible accurate and robust 6dof relative camera pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19854–19864, 2024.

[50] Jaroslav Moravec, Radim Šára, and Akihiro Sugimoto. Teso: Online tracking of essential matrix by stochastic optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28819–28827, 2026.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction describe the deployable virtual-camera formulation, image-wide behavior supervision, solver-aware observation calibration, and the evaluated rectification/downstream results. The claims are scoped to known-intrinsic online stereo rectification and the datasets used in the experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The discussion notes the known-intrinsics setting, the absence of dense ground-truth rectified correspondences in current online rectification benchmarks, and the need to evaluate alignment, downstream coverage, and metric-depth consistency together.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper proposes an empirical geometric learning and deployment framework and does not claim new theorems.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The method section specifies the optimization objectives, solver outputs, virtual-camera deployment, and coverage definition; the experiments specify datasets, resolutions, matchers, metrics, and training/testing settings that affect the reported results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper uses public datasets or public benchmark protocols and cites the corresponding sources. The authors will release when the it's fully organized.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experiments describe the data splits/protocols, image resolutions, stereo matchers, vertical-flow and keypoint metrics, downstream depth protocol, runtime measurement, and the two-stage training objective.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main VirtualRect results are averaged over five random seeds. Full confidence intervals are not included in the main text due to space, but the planned release includes per-sample logs so paired confidence intervals can be recomputed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The main paper reports runtime and a matcher/RANSAC/gauge breakdown on one NVIDIA A40 GPU. The release README will list hardware, checkpoint paths, and command-line configurations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses public or synthetic driving imagery and does not involve human-subject experiments, personal identification, or generative content.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The work targets a safety-relevant perception primitive for mobile robots and autonomous vehicles. The discussion emphasizes that rectification should be evaluated as a deployed stereo component rather than as a standalone row-alignment metric.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The work does not release generative models, scraped personal data, or other high-risk dual-use assets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites datasets, matchers, stereo networks, optical-flow models, robust estimators, and software libraries. The release README will list licenses and versions for reused code and pretrained checkpoints.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The new assets are code, trained checkpoints, configuration files, and evaluation scripts. They will be released with reproduction instructions when the paper is submitted.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects, so IRB or equivalent approval is not required.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not part of the proposed method, model training, or evaluation pipeline.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.