Computer Vision : Term Paper

Mayank Pratap
Purdue University
West Lafayette, USA
pratapm@purdue.edu

Abstract

Temporal consistency in depth estimation remains a critical challenge for video understanding, robotics, and AR/VR. Recent works approach this problem through diverse formulations, including deep temporal feature refinement, diffusion-based generative priors, and geometric point-cloud fusion. This paper provides a comparative critique of three representative methods: TC-Stereo [17], ChronoDepth [11], and Point-Based Fusion [5], analyzing their design assumptions, strengths, and limitations. We further implement and evaluate the point-based fusion method in an online monocular setting to assess its ability to balance temporal coherence and real-time performance. Our findings highlight trade-offs between accuracy, stability, and computational complexity, and suggest design directions for practical online depth estimation.

1. Introduction

Depth estimation from visual input is a fundamental problem in computer vision, with applications in robotics, autonomous navigation, augmented reality, and 3D reconstruction. While recent deep learning methods have achieved remarkable accuracy in single-image and stereo depth prediction, most of these approaches operate in a frame-independent manner. As a result, their per-frame predictions often suffer from significant temporal flicker, inconsistency, and geometric drift when applied to video sequences. Achieving *temporally consistent* depth estimation—producing coherent geometry across time while preserving spatial detail—remains a key challenge for video understanding systems.

Prior research has explored this problem from diverse perspectives. Feature-based approaches exploit temporal recurrence or cost-volume fusion between adjacent frames to enforce smooth disparity transitions. Generative approaches leverage video diffusion priors to learn implicit temporal correlations directly from data. Geometric methods, in contrast, focus on explicit 3D reasoning by maintaining temporally fused point clouds or voxel structures that integrate information over time. Each of these paradigms makes different assumptions about motion, scene dynamics, and computational trade-offs, leading to complementary strengths and limitations.

In this paper, we conduct a comparative study of three representative frameworks that capture these distinct philosophies: **Temporally Consistent Stereo Matching** (Zeng *et al.*, 2024) [17], a deep recurrent model that refines disparity maps through temporal state fusion; **ChronoDepth** (Shao *et al.*, 2025) [11], which employs diffusion-based generative priors to achieve long-range temporal stability; and **Point-Based Online Depth Fusion** (Khan *et al.*, 2023) [5], a geometric fusion method designed for real-time online operation. We critically analyze their architectures, training objectives, and assumptions regarding temporal modeling and scene dynamics.

Building on these insights, we implement and evaluate the point-based fusion framework, adapting it to our dataset and experimental setup. This implementation serves as a non-trivial codebase for exploring the practical trade-offs between accuracy, temporal coherence, and computational efficiency. Through quantitative metrics and qualitative visualizations, we assess how effectively this approach maintains temporal consistency under realistic scene motion. Our results highlight the ongoing need to bridge the gap between deep temporal reasoning and explicit geometric consistency for robust, real-time video depth estimation.

2. Background and Comparative Analysis

The goal of temporally consistent depth estimation is to maintain coherent depth predictions across consecutive video frames, mitigating flicker and geometric drift while preserving per-frame spatial accuracy. Recent research has approached this challenge through three complementary perspectives: (1) learning temporal dependencies within neural architectures, (2) leveraging diffusion-based generative priors for video stability, and (3) employing explicit 3D fusion strategies that integrate information over time. In this section, we review and critically analyze one represen-

tative work from each category, focusing on their key design choices, contributions, and limitations.

2.1. Temporally Consistent Stereo Matching

Zeng *et al.*propose **TC-Stereo** [17], a framework that extends stereo matching into the temporal domain by introducing temporal feature fusion and disparity refinement. The method consists of three main components: (1) *Temporal Disparity Completion* (TDC), which projects disparity maps from previous frames into the current view using known camera poses; (2) *Temporal State Fusion*, which merges historical and current features through a recurrent gating mechanism; and (3) *Dual-Space Refinement*, which jointly optimizes disparity and its spatial gradients for smoothness and edge preservation.

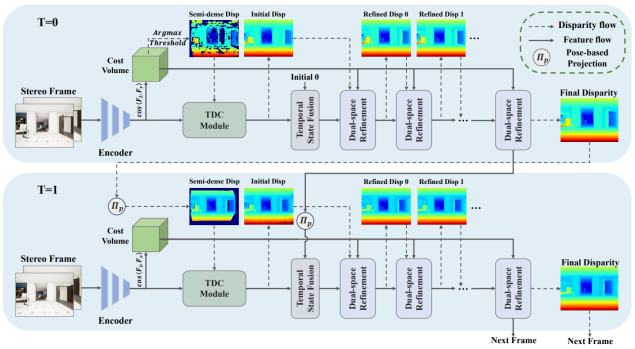

Figure 1. Architecture of the TC-Stereo framework proposed by Zeng et al. [17] The network integrates temporal information from consecutive stereo pairs to achieve temporally stable disparity estimation.

Figure 1 illustrates the architecture of the Temporally Consistent Stereo Matching (TC-Stereo) framework proposed by Zeng et al. [17]. The model introduces three primary components that together enforce temporal coherence in stereo depth estimation. The Temporal Disparity Completion (TDC) module projects disparity estimates from the previous frame into the current view, providing a well-initialized prediction. Next, the Temporal State Fusion block employs a gated recurrent unit (GRU) to merge historical and current features, capturing short-term temporal dependencies. Finally, the Dual-Space Refinement stage jointly optimizes the disparity map and its gradient field, improving edge sharpness and mitigating flicker between frames. This combination allows TC-Stereo to leverage both spatial and temporal cues efficiently while maintaining real-time performance. The training objective in TC-Stereo is defined as a combination of three components: a cost-volume loss $\mathcal{L}_{cv}$, a disparity loss $\mathcal{L}_{disp}$, and a disparity-gradient loss $\mathcal{L}_{grad}$ [17]. As described in Section 3.4 of their paper, the overall loss is formulated as:

$$\mathcal{L} = \mathcal{L}_{cv} + \mathcal{L}_{disp} + \mathcal{L}_{grad}. \quad (1)$$

Table 1. Benchmark comparison on KITTI-2015 (lower is better). TC-Stereo achieves both the highest accuracy and real-time speed.

Method	D1-all (%)↓	Time (s)↓
RAFT-Stereo [7]	1.82	0.38
CREStereo [6]	1.69	0.41
IGEV-Stereo [16]	1.59	0.18
TC-Stereo [17]	**1.46**	**0.09**

The cost-volume term $\mathcal{L}_{cv}$ supervises the similarity volume for sub-pixel disparity values following the HITNet formulation [13]. The disparity loss $\mathcal{L}_{disp}$ aggregates three sub-losses: disparity-completion, gradient-space refinement, and gradient-guided disparity-propagation, each penalizing the L_1 distance between predicted and ground-truth disparities at different refinement stages. The disparity-gradient loss $\mathcal{L}_{grad}$ enforces edge-aware consistency by aligning the gradients of the refined disparity map with the gradients of the ground truth. Together, these components encourage the model to recover geometrically sharp and temporally consistent disparity maps.

Performance is evaluated using the **D1-all** metric on the KITTI 2015 benchmark, which measures the percentage of pixels whose disparity error exceeds either 3 px or 5% of the ground-truth value, as seen in table 1. Overall, TC-Stereo presents a well-engineered approach for enforcing temporal consistency in stereo video. However, its reliance on accurate pose estimation and static-scene assumptions limits performance in dynamic environments. The recurrent fusion also increases memory usage compared to purely feed-forward stereo models.

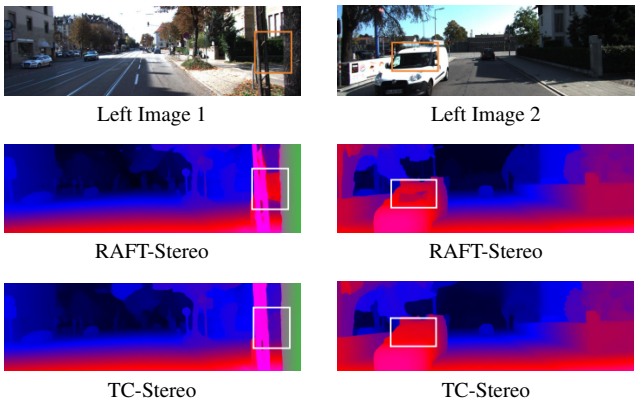

Figure 2. Qualitative comparison between RAFT-Stereo and TC-Stereo [17] on the KITTI 2015 dataset. Each column shows a different scene. The top row contains the left stereo inputs, the middle row shows the RAFT-Stereo predictions, and the bottom row shows results from TC-Stereo, which exhibit sharper boundaries and reduced temporal flicker.

2.2. Learning Temporally Consistent Video Depth from Video Diffusion Priors

Shao et al. propose **ChronoDepth** [11] , a diffusion-based video depth estimator designed to produce spatially accurate and temporally consistent depth for arbitrarily long videos. ChronoDepth introduces mechanisms to share context within and across video clips, enabling strong temporal coherence during streamed inference.

The authors reformulate depth prediction as a conditional denoising generation problem. During training, the model receives an RGB video clip and a corresponding latent depth representation, where distinct Gaussian noise levels are independently sampled for each frame. The denoiser learns to restore clean depth latents by DSM (denoising score matching) within the EDM framework [4]. To ensure temporal stability across clips, the authors propose a *consistent context-aware inference* strategy that initializes overlapping frames with previously predicted depth latents without adding noise, providing stable cross-clip conditioning.

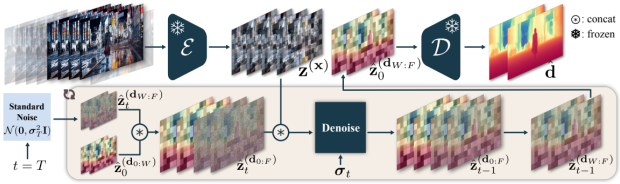

Figure 3. Overview of ChronoDepth's [11] context-aware inference pipeline. During streamed video processing, overlapping frames are initialized from previously predicted depth without noise, ensuring stable contextual guidance and significantly reducing cross-clip flicker.

Figure 3 illustrates the overall inference pipeline and how contextual information is propagated between clips. The underlying backbone is a UNet-like diffusion model with spatial and temporal layers. Spatial layers are trained first, followed by temporal layers trained on variable length clips.

Quantitative comparisons are reported in Table 2. The metrics used in these benchmarks include **AbsRel**, the absolute relative depth error; $\boldsymbol{\delta_1}$, the percentage of pixels whose depth prediction is within a 25% relative threshold of the ground truth; and **MFC**, the multi-frame consistency metric, which measures temporal flicker by comparing consecutive predicted depth maps. Lower AbsRel and MFC indicate better spatial accuracy and temporal stability, respectively, while higher δ_1 indicates more accurate depth recovery.

ChronoDepth achieves high temporal consistency on KITTI-360, ScanNet++, and Sintel, outperforming both single-image depth predictors and video-based approaches such as DepthCrafter and NVDS. Notably, it reduces multi-frame inconsistency (MFC) by up to 68% on KITTI-360 while maintaining competitive spatial accuracy.

Table 2. Comparison of ChronoDepth with representative baselines on the KITTI-360 benchmark. Lower AbsRel and MFC indicate better spatial accuracy and temporal stability; higher δ_1 is better.

Method	AbsRel↓	δ_1↑	MFC↓
Marigold [14]	0.213	**0.665**	0.776
DepthAnything V2 [2]	**0.207**	0.656	0.807
NVDS [15]	0.379	0.384	1.276
DepthCrafter [3]	0.293	0.462	0.655
ChronoDepth	0.215	0.654	**0.407**

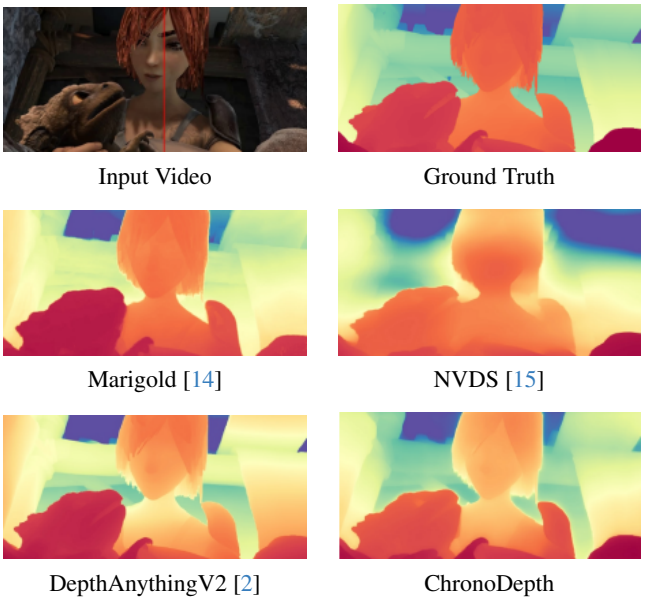

Figure 4. Qualitative comparison on the Sintel benchmark. The first row shows the input RGB frame and corresponding ground truth depth. The second and third rows compare representative baselines against ChronoDepth.

ChronoDepth effectively integrates temporal information through its context-aware inference mechanism, substantially reducing multi-frame flicker and enabling stable depth prediction on arbitrarily long videos. It achieves strong temporal consistency while maintaining competitive spatial accuracy, outperforming prior video depth methods such as NVDS and DepthCrafter. Its main limitations stem from the computational cost of diffusion sampling and its dependence on clip overlap; performance can degrade under rapid scene changes where temporal continuity breaks. Overall, ChronoDepth provides a clean and powerful framework for temporally consistent video depth, though its high inference cost restricts real-time or resource-limited deployment

2.3. Temporally Consistent Online Depth Estimation Using Point-Based Fusion

Khan et al. propose a lightweight online depth estimation method that enforces temporal consistency by fusing depth maps into a persistent 3D point-based representation [5]. The method is designed for streaming video, where each incoming frame is processed once, fused incrementally, and reprojected into future frames to provide stable depth estimates without the need for heavy recurrent models or diffusion sampling.

The system consists of three primary modules: a Point Cloud Update (PCU) module, a Temporal Fusion Network (TFN), and a Spatial Fusion Network (SFN).

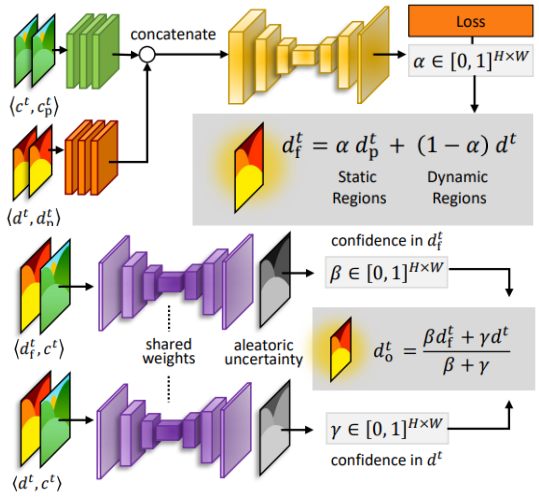

Figure 5. Overview of the two fusion modules in Khan *et al.* [5]. (Top) Temporal Fusion Network (TFN), which blends the reprojected prior depth with the current-frame depth using a learned blending weight α. (Bottom) Spatial Fusion Network (SFN), which applies uncertainty-aware smoothing to refine geometric consistency.

The PCU maintains a global fused point cloud by backprojecting depth predictions using camera intrinsics and transforming them with frame-to-frame poses. New points are merged using confidence-weighted updates and geometric pruning, producing a temporally stable 3D structure. This structure is then reprojected into the current frame to produce a temporal depth prior. The TFN takes as input the raw depth prediction and the reprojected temporal prior, and learns to correct temporal artifacts such as flicker and unstable depth jumps. Finally, the SFN refines the TFN output to correct spatial inconsistencies, sharpen depth boundaries, and suppress noise accumulated through long-term point fusion. Together, these modules enforce both temporal and spatial consistency while remaining suitable for real-time online processing.

The temporal and spatial fusion modules are trained using a combination of reconstruction, gradient, perceptual, and uncertainty-based losses. The blending weight α, responsible for selecting between the reprojected prior depth and the newly predicted depth, is encouraged to be binary through a BCE loss,

$$L_{\text{BCE}} = \text{BCE}\Big(\alpha,\ \mathbf{1}\Big(\|d_t^f - g_t\|_1 < \|d_t^p - g_t\|_1\Big)\Big). \quad (2)$$

The overall temporal fusion loss combines L1 depth supervision, the BCE regularizer, a gradient loss L_G, and a VGG-based perceptual loss:

$$L_\Theta = \|d_t^f - g_t\|_1 + L_{\text{BCE}} + L_G + \text{VGG}(d_t^f, g_t). \quad (3)$$

To improve robustness under motion blur or low-texture regions, the authors train the uncertainty network with an aleatoric uncertainty objective,

$$L_\Phi = \frac{1}{m}\left[\exp(-s^t)\,\|d_t^f - g_t\|_1 + \lambda_s\, s^t\right], \quad (4)$$

where s^t denotes the predicted log-uncertainty. d_t^f: fused depth prediction at time t; d_t^p: reprojected prior-frame depth; g_t: ground-truth depth at time t; α: learned blending weight between prior and fused depths; L_G: L1 loss on depth gradients; $\text{VGG}(\cdot)$: perceptual loss computed using VGG features; s^t: predicted log-uncertainty for frame t; m: number of valid pixels; λ_s: weighting term for the uncertainty regularizer. Together, these losses promote sharp geometry, stable temporal blending, and increased reliability in challenging image regions.

Table 3. Comparison on the MPI Sintel dataset [1] using key spatial and consistency metrics. Lower SC, SD(L1), RAE, and RMS indicate better depth accuracy and smoothness; higher δ_1 is better.

Method	SC↓	SD(L1)↓	AbsRel↓	RMS↓	δ_1↑
MiDaS [9]	0.675	0.597	0.279	3.013	0.610
WSVD [12]	0.704	0.654	0.360	3.775	0.466
DPT [10]	0.493	0.539	0.224	2.678	0.686
Authors-DPT	**0.295**	**0.474**	**0.197**	**2.400**	**0.710**

SC is the Spatial Consistency metric, which measures frame-to-frame gradient variation; lower values indicate smoother temporal transitions. SD(L1) is the Scale-Insensitive Depth Error, L1 depth error after scale alignment, reflecting spatial accuracy independent of scale drift.

The proposed point-based fusion method preserves a temporally consistent 3D map by combining reprojected prior depth with new depth estimates using learned blending weights. This design yields improved temporal consistency and sharper scene geometry compared to single-frame baselines such as DPT. The approach is lightweight and well-suited for online streaming, since it avoids costly diffusion-based sampling. However, its accuracy depends on reliable

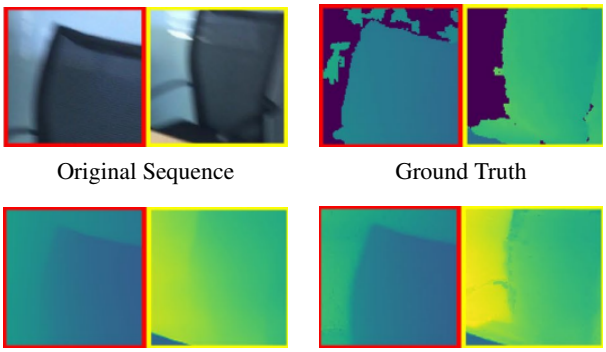

Figure 6. Qualitative comparison on the selected sequence. The authors' fused approach preserves sharper geometry and improves temporal consistency compared to the DPT baseline.

camera poses for reprojection, and the system may struggle in highly dynamic scenes where past depth becomes unreliable. Additionally, the dual fusion stages introduce sensitivity to calibration errors and accumulated point-cloud drift over long sequences.

3. Implementation

Our implementation is inspired by the framework proposed by *Khan et al.* [5], which comprises a temporal fusion network, a spatial fusion network, and a point-based global fusion module. While we follow the core design philosophy of the original method, several adaptations are necessary due to hardware and dataset constraints. The authors train on large-scale datasets such as FlyingThings3D [8] (200+ GB) and Sintel [1], using a cluster of six NVIDIA Tesla A100 GPUs. In contrast, our implementation is developed and trained on a single RTX 4060 Laptop GPU. To make the system feasible under these limitations, we adopt simplified network architectures, reduced training resolutions, and a more lightweight training protocol, while preserving the central ideas of temporal blending, uncertainty estimation, and point-cloud–based global fusion.

3.1. Point Cloud Update (PCU)

The Point Cloud Update module maintains a global 3D map that accumulates depth information across frames. This component is inspired by the point-based fusion strategy in Khan et al. [5], but we implement a lightweight version suitable for real-time operation on a single RTX 4060 Laptop GPU. The PCU performs three operations each frame: (1) unprojecting the current depth map into 3D, (2) projecting the existing global point cloud back into the image plane for temporal fusion, and (3) inserting new points in dynamic regions while maintaining a fixed memory footprint.

Given an RGB image c_t, its depth d_t, camera intrinsics K, and a predicted confidence map s_t, we convert each pixel (u, v) into a 3D point by standard pinhole unprojection:

$$x = (u - c_x)\frac{d_t(u,v)}{f_x}, \tag{5}$$

$$y = (v - c_y)\frac{d_t(u,v)}{f_y}, \tag{6}$$

$$z = d_t(u,v) \tag{7}$$

where (f_x, f_y, c_x, c_y) are the focal lengths and principal point extracted from K. All pixels with invalid depth or low confidence are discarded. The resulting positions, colors, and confidences are stored as:

$$\mathcal{P}_0 = \{(p_i,\, c_i,\, \sigma_i)\}_{i=1}^{N}. \tag{8}$$

To integrate temporally fused depth into the network, the global point cloud is reprojected into the current camera frame using the extrinsic matrix T,

$$p_i^{\text{cam}} = T \begin{bmatrix} p_i \\ 1 \end{bmatrix}, \tag{9}$$

followed by camera projection,

$$u = f_x \frac{x_{\text{cam}}}{z_{\text{cam}}} + c_x, \qquad v = f_y \frac{y_{\text{cam}}}{z_{\text{cam}}} + c_y. \tag{10}$$

We apply vectorized z-buffering via scatter operations, keeping only the closest point for each pixel. This produces a pseudo-depth map d_t^{proj} and projected colors c_t^{proj}, which are used as prior information during temporal fusion.

After obtaining the fused depth d_t^f and the temporal blending mask α_t from the Temporal Fusion Network, we selectively insert new 3D points only where temporal fusion indicates that current-frame depth should be trusted:

$$M_t = \big(\alpha_t \geq 0.5\big) \wedge \big(d_t^f > 0\big) \wedge \big(s_t > \tau\big), \tag{11}$$

where s_t is the predicted confidence and τ is a predefined threshold. Pixels satisfying M_t are unprojected into 3D and appended to the global point cloud:

$$\mathcal{P}_{t+1}\mathcal{P}_t \cup \left\{ \big(p(u,v),\, c_t(u,v),\, s_t(u,v)\big) \;\middle|\; (u,v) \in M_t \right\}. \tag{12}$$

To remain memory-efficient, we keep only the top-K most confident points when the cloud exceeds a fixed size (set to 500,000 points in our implementation).

The PCU acts as a temporal accumulator of 3D geometry. It enables the network to:

- maintain long-range 3D consistency across frames,
- provide a geometric prior d_t^{proj} for the temporal fusion network,

- preserve details that are stable across multiple frames, and
- avoid drift by downweighting low-confidence points.

While the original method in [5] uses heavier GPU-based fusion operations and high-resolution global point maps, our implementation introduces a simplified, fully vectorized version that remains real-time on a single consumer GPU, enabling practical experimentation under limited hardware.

3.2. Temporal Fusion Network

The Temporal Fusion Network (TFN) is responsible for merging the current depth estimate with a reprojected prior depth map from the previous frame. Following the formulation of Khan et al. [5], the TFN predicts a pixelwise blending weight $\alpha_t \in [0,1]^{H\times W}$, which selects between trusting the current depth prediction or the temporally propagated prior depth.

In our implementation, the TFN is a lightweight encoder–decoder with three convolutional layers for downsampling and two transposed-convolution layers for upsampling. The input consists of the concatenation of the current and previous RGB frames (c_t, c_{t-1}) and their corresponding depth maps (d_t, d_{t-1}), forming an 8-channel tensor:

$$x_t = [c_t, c_{t-1}, d_t, d_{t-1}] \in \mathbb{R}^{8\times H\times W}.$$

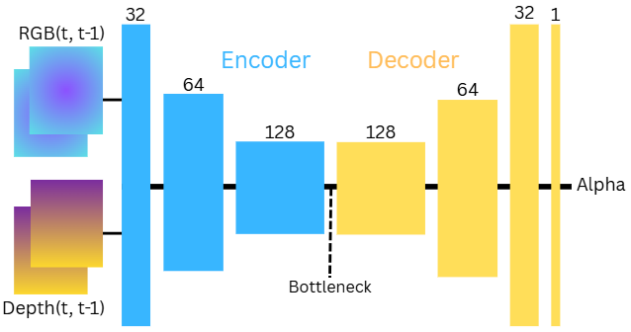

Figure 7. Architecture of the Temporal Fusion Network (TFN). The model receives the current and previous RGB frames and depth maps, concatenated into an 8-channel input tensor. The encoder extracts multi-scale features through three convolutional blocks with channel sizes {32, 64, 128}, reducing spatial resolution via strided convolutions. A decoder with mirrored transposed convolutions upsamples the features and produces a single-channel fusion mask $\alpha_t \in [0,1]$. This mask determines the per-pixel weighting between the current depth estimate and the reprojected prior depth, enabling the model to learn when to trust temporal history versus new observations

After spatial encoding and decoding, the network outputs a single-channel fusion weight map:

$$\alpha_t = \sigma\left(\frac{\Psi(x_t)}{T}\right), \tag{13}$$

where Ψ denotes the decoder output and T is a temperature parameter ($T = 1$ during training). The final fused depth is given by the convex combination:

$$d_t^f = \alpha_t\, d_t + (1-\alpha_t)\, d_{t-1}. \tag{14}$$

This explicit formulation allows the TFN to learn when temporal consistency is reliable and when newly predicted depth should dominate, particularly in dynamic or fast-moving regions.

The temporal fusion network is supervised using a combination of reconstruction, classification, gradient, and perceptual losses. The fused depth prediction d_t^f is encouraged to match the ground-truth depth g_t through an L1 reconstruction term,

$$L_{\mathrm{L1}} = \|d_t^f - g_t\|_1. \tag{15}$$

To ensure that the fusion mask α_t behaves as a binary selector between the current-frame depth and the reprojected prior, the model is trained with a binary cross-entropy loss,

$$L_{\mathrm{BCE}} = \mathrm{BCE}(\alpha_t,\, \alpha_t^{\mathrm{gt}}), \tag{16}$$

where α_t^{gt} is generated from synthetic motion masks and depth-difference heuristics as described in Khan et al.

To promote sharp geometric boundaries, a gradient-matching term is added,

$$L_{\mathrm{grad}} = \left\|\nabla_x d_t^f - \nabla_x g_t\right\|_1 + \left\|\nabla_y d_t^f - \nabla_y g_t\right\|_1. \tag{17}$$

A VGG-based perceptual loss further encourages depth structures to align with higher-level features in the ground-truth depth map,

$$L_{\mathrm{VGG}} = \sum_i \left\|\phi_i(d_t^f) - \phi_i(g_t)\right\|_1, \tag{18}$$

where ϕ_i denotes selected layers of a pretrained VGG-16 feature extractor.

The complete temporal fusion objective combines these components with fixed weights,

$$L_\Theta = 4.0\, L_{\mathrm{L1}} + 8.0\, L_{\mathrm{BCE}} + 0.05\, L_{\mathrm{grad}} + 0.05\, L_{\mathrm{VGG}}. \tag{19}$$

This formulation closely follows the loss design in Khan et al. [5], with minor modifications to improve training stability on limited hardware.

3.3. Spatial Fusion Network

The Spatial Fusion Network (SFN) serves as the uncertainty estimation module in our implementation, following the formulation of Khan et al. [5]. Given the RGB frame c_t and an input depth map (either d_t or the fused prediction d_t^f), the

network predicts an aleatoric uncertainty map s^t, which is later used to modulate the depth fusion process.

Our SFN is a U-Net–style encoder–decoder architecture with instance normalization and ReLU activations throughout. The input tensor $[c_t, d_t] \in \mathbb{R}^{4\times H\times W}$ is first processed through four encoder stages with progressively increasing channel width $(32, 64, 128, 256)$ and spatial downsampling via strided convolutions. The decoder mirrors this structure through transposed convolutions, and skip connections are used to preserve spatial detail. The final 1×1 convolution outputs a single-channel log-variance prediction:

$$s^t = \Phi(c_t, d_t) \in \mathbb{R}^{1\times H\times W}.$$

This design closely follows the uncertainty prediction module described in Khan et al. [5], while being scaled down to accommodate our hardware constraints.

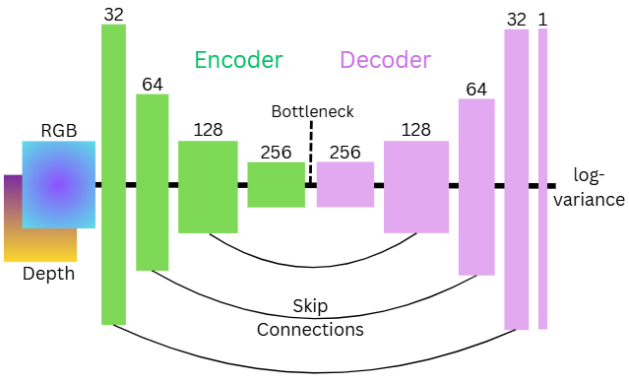

Figure 8. Architecture of the Spatial Fusion Network (SFN). The input RGB image and predicted depth are concatenated into a 4-channel tensor and processed through a four-level encoder with channel sizes {32, 64, 128, 256}. Each encoder block uses Conv–IN–ReLU layers and progressively downsamples the resolution. The decoder mirrors this structure with transposed convolutions and skip connections that concatenate encoder features at corresponding scales. The final 1-channel output predicts the log-variance (aleatoric uncertainty) associated with each depth pixel, which is later used to compute confidence weights during spatial fusion

Training the SFN requires encouraging low uncertainty when depth predictions are accurate, and higher uncertainty in ambiguous or noisy regions. Following our implementation, we adopt an edge-aware uncertainty loss composed of four terms: a data term, a regularization term, a gradient-consistency loss, and an edge-confidence penalty.

Given the predicted uncertainty s^t, the DPT depth estimate d_t^{DPT}, and the ground-truth depth g_t, the residual is $r_t = |d_t^{\mathrm{DPT}} - g_t|$. The base uncertainty objective is:

$$L_{\mathrm{data}} = \frac{1}{m}\sum_i \exp(-s_i^t)\, r_{t,i}, \qquad L_{\mathrm{reg}} = \frac{1}{m}\sum_i s_i^t, \quad (20)$$

where m is the number of pixels. To preserve structural edges, we compute horizontal and vertical depth gradients and minimize their L1 difference:

$$L_{\mathrm{grad}} = \left\|\nabla_x d_t^{\mathrm{DPT}} - \nabla_x g_t\right\|_1 + \left\|\nabla_y d_t^{\mathrm{DPT}} - \nabla_y g_t\right\|_1. \quad (21)$$

Finally, an edge-confidence loss pushes the network to assign higher uncertainty in regions of large ground-truth gradient magnitude, where depth ambiguity is naturally greater:

$$L_{\mathrm{edge}} = \frac{1}{m}\sum_i s_i^t \cdot \mathbf{1}(|\nabla g_t|_i > \mathrm{mean}(|\nabla g_t|)). \quad (22)$$

The total spatial fusion loss is:

$$L_{\mathrm{SFN}} = L_{\mathrm{data}} + \lambda_{\mathrm{reg}} L_{\mathrm{reg}} + \lambda_{\mathrm{grad}} L_{\mathrm{grad}} + \lambda_{\mathrm{edge}} L_{\mathrm{edge}}, \quad (23)$$

with weighting coefficients $\lambda_{\mathrm{reg}} = 0.03$, $\lambda_{\mathrm{grad}} = 0.5$, and $\lambda_{\mathrm{edge}} = 0.05$ in our implementation.

4. Training and Evaluation

Our full system follows the three-stage pipeline introduced by Khan et al. [5], consisting of (1) an image-based depth estimator, (2) a temporal fusion network, (3) a spatial fusion network for uncertainty-aware refinement, and (4) a global point-cloud fusion module for long-term temporal consistency. We train the Spatial and Temporal Fusion Networks on the Sintel training dataset, for 10 epochs each. During inference, each incoming RGB frame is first processed by a pretrained DPT model to obtain an initial depth estimate. For the first frame of each sequence, this depth, together with the RGB image, is used to initialize a global 3D point cloud. For subsequent frames, the point cloud is reprojected into the current view to provide a geometric prior. The temporal fusion network blends this prior with the new DPT depth using a learned fusion mask α_t, followed by spatial refinement with an uncertainty map produced by the spatial fusion network. The refined depth map is then used to update the global point cloud, enabling consistent geometry across long sequences.

Table 4. Comparison of our implementation with the authors' method (Ours-DPT [5]) on the MPI-Sintel dataset [1]. Lower AbsRel, RMSE, SC, and SD(L1) indicate better accuracy; higher δ_1 is better.

Method	AbsRel↓	RMSE↓	SC↓	SD(L1)↓	δ_1↑
Authors [5]	**0.255**	**2.400**	**0.295**	**0.474**	**0.710**
Ours	0.389	3.316	0.493	0.616	0.532

Our reimplementation achieves the same qualitative behavior as the authors' method but underperforms quantitatively for several reasons. First, we train on a significantly

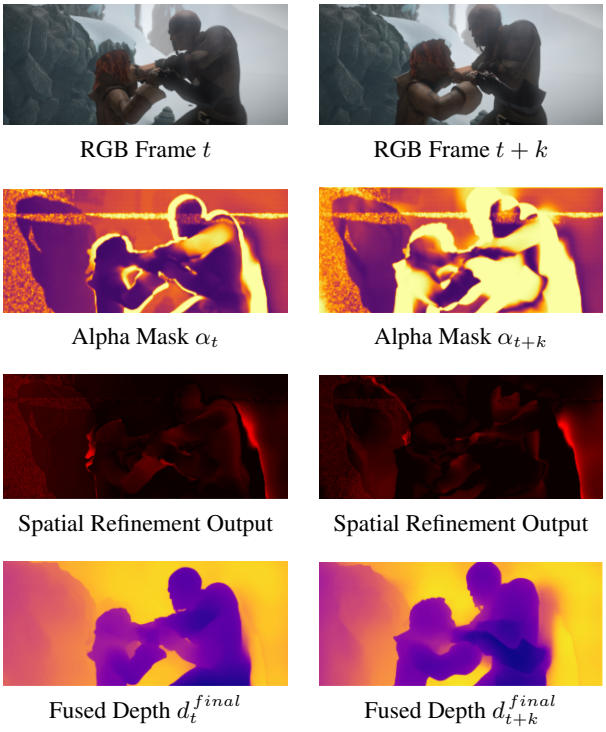

Figure 9. Visualization of our temporal–spatial fusion pipeline on two Sintel frames. Row 1 shows the RGB inputs; Row 2 shows the learned temporal fusion masks α (darker regions trust the previous frame, lighter regions trust the current frame); Row 3 shows the spatial refinement outputs, which sharpen geometry and suppress noise; and Row 4 shows the final fused depth maps. Together, these stages combine temporal priors with spatial uncertainty cues to produce stable and sharp depth estimates across time.

smaller dataset (only the Sintel training split) compared to the large-scale FlyingThings3D + Sintel combination used in the original work, limiting the model's ability to generalize scene geometry and motion patterns. Second, our networks use reduced architectural capacity to accommodate a single RTX 4060 laptop GPU, whereas the authors train deeper models on six A100 GPUs. Third, our training schedule is necessarily shorter: fewer epochs, lower resolution, and smaller batch sizes, which restricts convergence and refinement of the fusion weights and uncertainty estimates. Together, these constraints naturally lead to higher reconstruction error and lower temporal consistency compared to the fully trained, large-scale system reported in the original paper.

4.1. Future Work

Several directions can further improve the performance and robustness of our system. First, training on a broader set of datasets, including FlyingThings3D, KITTI, and ScanNet—would expose the model to a wider range of motions, geometries, and lighting conditions, reducing overfitting to Sintel and improving generalization. Second, exploring more efficient network architectures, such as lightweight UNet variants, depthwise-separable convolutions, or transformer-based fusion modules, may provide better accuracy–efficiency tradeoffs suitable for real-time operation. Finally, incorporating more advanced point-cloud representations (e.g., voxel hashing or sparse convolutional backbones) could further stabilize global fusion while reducing computational overhead.

References

[1] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV)*, pages 611–625. Springer-Verlag, 2012. 4, 5, 7

[2] Daniele Rege Cambrin, Isaac Corley, and Paolo Garza. Depth any canopy: Leveraging depth foundation models for canopy height estimation. *arXiv preprint arXiv:2408.04523*, 2024. 3

[3] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2005–2015, 2025. 3

[4] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, pages 26565–26577. Curran Associates, Inc., 2022. 3

[5] Numair Khan, Eric Penner, Douglas Lanman, and Lei Xiao. Temporally consistent online depth estimation using point-based fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9119–9129, 2023. 1, 4, 5, 6, 7

[6] 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*, pages 16263–16272, 2022. 2

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

[8] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134. 5

[9] 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*, 44(3):1623–1637, 2020. 4

[10] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of*

the IEEE/CVF international conference on computer vision, pages 12179–12188, 2021. 4

[11] Jiahao Shao, Yuanbo Yang, Hongyu Zhou, Youmin Zhang, Yujun Shen, Vitor Guizilini, Yue Wang, Matteo Poggi, and Yiyi Liao. Learning temporally consistent video depth from video diffusion priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22841–22852, 2025. 1, 3

[12] Feitong Tan, Hao Zhu, Zhaopeng Cui, Siyu Zhu, Marc Pollefeys, and Ping Tan. Self-supervised human depth estimation from monocular videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 650–659, 2020. 4

[13] 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*, pages 14362–14372, 2021. 2

[14] Massimiliano Viola, Kevin Qu, Nando Metzger, Bingxin Ke, Alexander Becker, Konrad Schindler, and Anton Obukhov. Marigold-dc: Zero-shot monocular depth completion with guided diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5359–5370, 2025. 3

[15] Yiran Wang, Min Shi, Jiaqi Li, Zihao Huang, Zhiguo Cao, Jianming Zhang, Ke Xian, and Guosheng Lin. Neural video depth stabilizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9466–9476, 2023. 3

[16] 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*, pages 21919–21928, 2023. 2

[17] Jiaxi Zeng, Chengtang Yao, Yuwei Wu, and Yunde Jia. Temporally consistent stereo matching. In *European Conference on Computer Vision*, pages 341–359. Springer, 2024. 1, 2