TemporalStereo: Efficient Spatial-Temporal Stereo Matching Network

Youmin Zhang Matteo Poggi Stefano Mattoccia
Department of Computer Science and Engineering (DISI), University of Bologna
https://youmi-zym.github.io/projects/TemporalStereo/

Abstract—We present TemporalStereo, a coarse-to-fine stereo matching network that is highly efficient, and able to effectively exploit the past geometry and context information to boost matching accuracy. Our network leverages sparse cost volume and proves to be effective when a single stereo pair is given. However, its peculiar ability to use spatio-temporal information across stereo sequences allows TemporalStereo to alleviate problems such as occlusions and reflective regions while enjoying high efficiency also in this latter case. Notably, our model – trained once with stereo videos – can run in both single-pair and temporal modes seamlessly. Experiments show that our network relying on camera motion is robust even to dynamic objects when running on videos. We validate TemporalStereo through extensive experiments on synthetic (SceneFlow, TartanAir) and real (KITTI 2012, KITTI 2015) datasets. Our model achieves state-of-the-art performance on any of these datasets.

I. INTRODUCTION

Stereo reconstruction is a fundamental problem in computer vision. It aims at recovering the 3D geometry of the sensed scene by computing the disparity between corresponding pixels in a rectified pair of images. In robotic [34], virtual/augmented reality [54], autonomous driving [37], and more, depth information is crucial for scene understanding, with accurate yet fast solutions being particularly attractive.

Recently, deep learning approaches have shown tremendous improvements, especially in the supervised setting [49], [7], [21]. Common to most architectures is the cost volume, which encodes the feature similarity and plays a key role in matching pixels. In particular, features from the left image are concatenated (in 3D networks) or correlated (in 2D networks) and processed through convolutions, which are costly and prevent real-time execution. To address this issue, more efficient stereo models have been proposed, taking advantage of efficient cost aggregation [3], [50], coarse-to-fine [19], [47] and cost volume pruning [8], [26], [36] strategies to preserve computation and accuracy. Nonetheless, the sparse and constant disparity candidates used for matching cost computation can easily miss the ground truth disparity, thus deteriorating the accuracy of prediction. The case becomes even worse when dealing with challenging regions. For example, although occlusion is an ill-posed problem for stereo matching based on a single stereo pair, popular supervised stereo methods have not ever taken advantage of the intrinsic correlation among frames when stereo videos are available. Even if commercial cameras can easily acquire high FPS videos (e.g., 30 or more), in which likely past context is strictly related to the current one, state-of-the-art stereo methods generally process each frame independently.

We argue that previous disparities can be relevant cues to improve estimates, especially in complex regions where the current pair alone is not sufficiently meaningful, such as near object boundaries and occlusions.

Inspired by these observations, we propose TemporalStereo, a novel lightweight deep stereo model (as depicted in Fig. 1) being not only suited for single pair fast inference, but also able to leverage the rich spatial-temporal cues provided by stereo videos, when available. In single-pair mode (i.e., when only the current pair is available), our network follows a coarse-to-fine scheme [47], [19], [12] to enable efficient inference. In order to increase the capacity of the constructed sparse cost volume and also the quality of sparse disparity candidates at every single stage, we enhance the cascade-based architecture in several aspects: 1) firstly, we enrich the cost of each queried disparity candidate with context from non-local neighborhoods during cost computation; 2) as the number of disparity candidates could be decreased to very few (e.g., 5), in addition to exploiting pure 3D convolutions for cost aggregation, we also perform multiple statistics (e.g., mean and maximum statistics) over a large window to grasp the global cost distribution of each pixel at one stage; 3)

Fig. 1: Overview of TemporalStereo. Our model can seamlessly process a stereo pair, as well as switch to temporal mode when multiple pairs are available, to increase accuracy. This allows TemporalStereo for achieving a favorable accuracy/speed trade-off with respect to existing networks. GMacs are measured with a resolution of 960x572.
differently from previous strategies [26], [19], [47], [36], for which the candidates are constant throughout each stage, we rely on the aggregated cost volume itself to shift the candidates towards a better location and thus improve the quality of predicted disparity map; 4) additionally, when multiple pairs are available, our network can easily switch to temporal mode, in which past disparities, costs and cached features could also be aligned to the current reference frame and used to boost current estimates with a negligible runtime increase (4ms as shown in Tab. VI). Experimental results on synthetic (SceneFlow, TartanAir) and real (KITTI 2012, KITTI 2015) benchmarks support the following claims:

- Our method is accurate yet fast and achieves state-of-the-art results when a single stereo pair is available. In particular, the improved coarse-to-fine design based on very few (e.g., 5) candidates allows TemporalStereo to unlock fast execution and high performance.

- When multiple temporally adjacent stereo pairs are available, TemporalStereo is the first supervised stereo network that can effortlessly cache the past context and use it to improve ongoing predictions, e.g., in occluded and reflective regions. The model trained in temporal mode is effective also in single-pair mode, allowing the deployment of the same model in both settings.

- The proposed temporal cues can be widely applied to boost the matching accuracy of current efficient stereo methods, e.g., on TartanAir [46] dataset, the improvements of CoEx [3] and StereoNet [19] are 14.6% and 26.1% respectively. Nonetheless, our TemporalStereo is still the best one for utilizing temporal information.

II. RELATED WORK

Single Pair Stereo Matching. Traditional methods [15] employ handcrafted schemes to find local correspondences [33], [53] and approximate global optimization by exploiting spatial context [14]. Recent deep learning strategies [31] adopt CNNs and can be broadly divided into two categories according to how they build the cost volume. On the one hand, 2D networks follow DispNetC [27] employing a correlation-based cost volume and applying 2D convolutions to regress a disparity map. Stacked refinement sub-networks [22] and multi-task learning [38], [52] have been proposed to improve the accuracy. GCNet [18] represents a milestone for 3D networks. Following works improve its results thanks to spatial pyramid pooling [6], group-wise correlations [13], forcing unimodal [57], [10] or bimodal [45] cost distributions. 3D methods usually outperform 2D architecture by a large margin on popular benchmarks [27], [11], [28], paying more in terms of computational requirement. In this work, we leverage 3D sparse cost volumes, and the peculiar proposed architecture proves to be fast and accurate at disparity estimation.

Efficient Stereo Matching with Deep-Learning. A popular strategy to decrease the runtime consists in performing computation at a single yet small resolution (e.g., 1/4) and obtaining the final disparity through upsampling. CoEx [3] adopts this strategy. Coarse-to-fine [2], [51] design further improves efficiency, since the overall computation is split into many stages. To further reduce the disparity search range for each pixel, StereoNet [19] and AnyNet [47] add a constant offset to disparity maps produced in the previous stage to avoid checking all the disparity candidates. However, as reported in [26], the constant offset strategy is not robust against large errors in initial disparity maps. DeepPruner [8] prunes the search space using a differentiable variant of PatchMatch [4], obtaining sparse cost volumes. In contrast, methods such as [26], [36] employ uncertainty to evaluate the outcome of previous stages and then build the current cost volume accordingly. However, since the cost volume itself expresses the goodness of candidates (the better the candidates, the more representative the cost volume), it could be used to check and eventually correct the candidates themselves. Nonetheless, previous methods in literature cannot exploit this cue since their candidates come from the earlier level and are constant through the current stage. Conversely, this paper proposes inferring an offset for each disparity candidate from the current aggregated cost volume, which helps to ameliorate the disparity. Moreover, to improve network efficiency on high-resolution images, we perform the coarse-to-fine predictions only at downsampled feature maps rather than at full image resolution like HITNet [40].

Multiple Pairs Stereo Matching. When two temporally adjacent stereo pairs are available, optical flow [27], [42] or 3D scene flow [44] is often taken into account to link the images with the reconstructed 2D/3D motion field. Then, the stereo problem is tackled by leveraging a multi-task model [1], [20], [48] or by casting it as a special case of optical flow [24]. However, these methods are not able to capitalize longer stereo sequences. Conversely, OpenStereoNet [58] adopts recurrent units to capture temporal dynamics and correlations available in videos for unsupervised disparity estimation. We argue that past information could be beneficial to understand the current scene – especially in difficult areas – for free since past context and predictions can be cached easily, although none of the above methods make use of them explicitly in videos by taking into account geometry. On the contrary, TemporalStereo fully exploits this potential.

III. METHOD

We first describe TemporalStereo in single-pair mode, then we fully unlock the potential of our architecture enabling temporal mode. Fig. 2 depicts our model.

A. Single Pair Mode

Given a single stereo pair, a backbone extracts multi-scale features at 1/4, 1/8, and 1/16 of the original resolution. Then, three stages predict disparity maps starting from these features. In particular, each stage performs feature decoding, cost volume computation, and aggregation.

Feature decoding In each stage \( s \in \{1, 2, 3\} \), a decoder processes the current features, together with those from the previous stage \( F_s^{s-1} \) if \( s > 1 \). In particular, \( F_1^{s-1} \) are bilinearly upsampled by a factor 2 and concatenated with left and right features from the backbone.
Then, the resulting feature maps are processed by two 2D convolutions with kernels of 3, obtaining $F_i^s$, $F_s^s$.

Cost volume computation. A 4D cost volume $C^s \in \mathbb{R}^{H_s \times W_s \times |D^s|}$ is constructed by concatenating $F_i^s$ with the corresponding $F_s^s$ for each disparity $d$ in the set $D^s = \{d_1, d_2, ..., d_n\}$, with $H_s$ respectively the number of channels of feature $F_i^s$ and disparity candidates used in the stage:

$$C_{\text{concat}}^s(\cdot, u, v, d) = \bigoplus \{F_i^s(\cdot, u, v), F_s^s(\cdot, u - d, v)\},$$

where $u, v$ are horizontal and vertical pixel coordinates, respectively, while $\bigoplus$ stands for concatenation on channel dimension. However, by doing so, $C^s$ only contains local information, which is a major limitation in the case of a sparse set of candidates. To alleviate it, we enrich $C^s$ with multi-level costs [42] encoding a larger neighborhood. Specifically, we build a three-level cost volume from feature maps $F_i^s$, $F_s^s$ and by downsampling the costs in its neighborhood, collecting matching results of nearby pixels. At each level, we perform group-wise correlations [13], then costs are bilinearly upsampled to $C^s$ resolution and stacked together, obtaining $C_{\text{agg}}^s$. The final multi-level cost is:

$$C^s(\cdot, u, v, d) = \bigoplus \{C_{\text{concat}}^s(\cdot, u, v, d), C_{\text{agg}}^s(\cdot, u, v, d)\}.$$

Cost aggregation. We now describe how we compute the aggregated cost volume $C_{\text{agg}}^s$. In each stage, preliminary spatial cost aggregation is performed by one 3D convolution with kernel size $3 \times 3 \times 3$ followed by an hourglass network and another 3D convolution with kernel size $3 \times 3 \times 3$. Afterward, a Statistical Fusion module further improves the aggregation with mean and maximum statistics [56], especially along the disparity dimension (which contains as few as 5 candidates): 4 parallel layers (identity, convolutional with kernel size $5 \times 1 \times 1$, average pooling and max pooling both with kernel size $5 \times 5 \times 5$, respectively) extract global statistics from the preliminary aggregated volume, then their outcomes are stacked together and processed by one 3D convolution. In temporal mode, Statistical Fusion also helps to merge past costs with current ones, as we will discuss later. We employ Statistical Fusion only in stages $s = 1, 2$ to lighten the computation, while in $s = 3$ past costs are not used. Finally, a cost prediction head with two 3D convolutions with kernel size $3 \times 3 \times 3$ is in charge of predicting the final cost volume $C_f^s$ from $C_{\text{agg}}^s$. From $C_f^s$ and $D^s$ we could obtain the disparity $d^s$ of the current stage by means of soft-argmin operator [18], but $C_{\text{agg}}^s$ can help us to improve the candidates – as we will describe thereafter. For this reason, before computing $d^s$, we have to introduce our adaptive candidate shifting strategy.

Adaptive candidate shifting. Previous works [26], [19] consider the candidates $D^s$ as constants throughout the current stage $s$. However, the aggregated cost volume itself contains cues about the candidates used to build it. Since it has been constructed from $D^s$, it represents how well the candidates can explain the current scene. Moreover, in the cost aggregation step, each cost has been enriched with all the costs in its neighborhood, collecting matching results of nearby pixels. We propose to leverage a dedicated prediction head to infer, from $C_{\text{agg}}^s$ as shown in Fig. 2, an offset value for each candidate, which represents how much we have to shift the candidate towards a better location. Thus, the candidates are pixel-wise different and able to adaptively adjust according to the learned context. This strategy gives the network the ability to adapt $D^s$ and improve $d^s$, as depicted in Fig. 3. Specifically, an independent head – with the same structure of the cost prediction head, i.e., two 3D convolutions, – is in charge of inferring the offset volume $O^s \in \mathbb{R}^{H_s \times W_s \times |D^s|}$ from $C_{\text{agg}}^s$ as input. Although offset learning [25] is not new and has been used in [10] to convert the dense disparity candidates, which always tightly cover the ground truth disparity, into continuous ones so that the predicted disparity value will not be restricted to integers, in contrast, our disparity candidates are few and sparse, which could easily miss the ground truth. By enriching the context of each sparse candidate, the proposed shifting strategy is able to correct the constant and flawed candidates for better disparity estimation.

Disparity prediction and candidate sampling. Given $D^s$, $O^s$ and $C_f^s$, soft-argmin operator [18] could be applied to predict the final disparity map. However, this strategy results sub-optimal in the case of multi-modal distributions [57], [10], [45]. To overcome this limitation, we predict the
disparity following the strategy proposed in [3]: we select top-K values of \(-C_s\) for each pixel across the dimension, and we normalize them with the softmax operator \(\sigma\). The predicted disparity map \(d^{\star}\) results:

\[
d^{\star} = \sum_{d \in \{d_1^{\star}, \ldots, d_K^{\star}\}} (d + o_d) \times \sigma(-c_d),
\]

where \(d_k^{\star} = \arg\max_k (c_d)\), \(\arg\max_k (\cdot)\) is the k-th maximal value for \(k \in \{1, 2, \ldots, K\}\), \(c_d\) the cost in \(C_d\) of candidate \(d\) for each pixel, and \(o_d\) is the offset value in \(O\) for the candidate \(d\). Convex upsampling [42] is used to double the resolution of \(d^{\star}\) and \(d^3\), obtaining \(d_1^{\star}, d_2^{\star}\). To bring \(d^{\star}\) to full resolution, following [3], we bilinearly upsample \(d^{\star}\) to full resolution and for each pixel in the upsampled resolution, a weighted average of a \(3 \times 3\) superpixel surrounding it is calculated to get the final prediction \(d^{\star}\). Finally, we obtain the set of candidates \(D^{s+1}\) for the next stage applying an inverse transform sampling. We sample according to a normal distribution \(N(d^{s+1}_j, 1)\) in the range \([-\beta, \beta + \beta]\), with \(\beta\) a constant value. This strategy allows to include more candidates near the predicted disparity. The first stage is initialized with \(D^1\) uniformly sampled across the full range to provide an overview of current scene geometry.

**Loss function.** Following [6], we minimise the smooth \(L_1\) loss, i.e., Huber loss at full resolution between the ground truth disparity map \(d^{gt}\) and the predictions at the three stages \(\{d_1^{\star}, d_2^{\star}, d_3^{\star}\}\). \(d_1^{\star}, d_2^{\star}\) and \(d^{\star}\) are bilinearly upsampled to full resolution and a factor \(\lambda\) weights each predicted map. The loss results:

\[
L_{huber} = \frac{1}{|N|} \cdot \text{smooth}_{L_1}(d^{gt} - d^{\star}) + \sum_{s=1}^{3} \frac{1}{|N|} \cdot \text{smooth}_{L_1}(d^{gt} - d^{s}),
\]

where \(N\) is the number of pixels with valid ground truth.

Furthermore, to supervise the learning of offsets in three stages \(\{O^1, O^2, O^3\}\), we adopt the Wasserstein loss [10] for training. Specifically, for each stage \(s\), we downsample the ground truth disparity map \(d^{gt}\) to the resolution of corresponding offset for supervision, obtaining \(d^{gt,s}\). As the offset value can range in a quite large value space, we improve the Wasserstein loss as:

\[
L_{war} = \sum_{s=1}^{3} \frac{1}{|N|} \cdot \sum_{d \in D_s} \left( |d + o_d - d^{s}\star| \times \sigma(-c_d + \alpha) \right),
\]

where \(c_d\) the cost value in \(C_d\) of candidate \(d\) for each pixel, and \(o_d\) is the offset value in \(O^s\) for the candidate \(d\). We set \(\alpha = 0.25\), i.e., even for the candidate with extremely low matching probability \(\sigma(-c_d)\), the network still learns an offset to enforce it moving towards the ground truth. With a weight factor \(\lambda_{final}\), our final loss becomes:

\[
L = L_{huber} + \lambda_{final} \cdot L_{war}.
\]

**B. Temporal Mode**

So far, we have presented the model suited for single-pair mode. Now, we illustrate how the model works in temporal mode. Differently from multi-view stereo models [16], [41], our network processes stereo videos one stereo pair at a time. This behaviour, which helps to save computation, is possible thanks to the sparse cost volume. In fact, we can easily add disparity candidates from the past inferences to the current set \(D^s\), thus increasing the search range with other plausible solutions. The past candidates, however, have to be aligned with the current frame to be meaningful. Optical flow/3D scene flow could be used to tackle the issue, but predicting flow fields is expensive in terms of memory footprint and time. Instead, in this work, we suppose that the camera is calibrated and the pose is given – since pose can be provided by external IMU sensors or estimated from the stereo pairs – to build the rigid motion field needed to align the candidates. Nonetheless, rigid flow formulation only holds for static objects in the scene, but since our network always relies on the current stereo pair, TemporalStereo is robust even in case of wrong flows (e.g., due to bad poses) or moving objects. In the remainder, we detail how we do use temporal information to improve stereo-matching results.

**Local map.** As highlighted before, occlusions represent a major issue in stereo matching. However, we argue that currently occluded regions might have been visible in past frames, e.g., due to camera motion. For this reason, the issue can be alleviated by adequately exploiting past estimates and at a minimal cost since they can be easily cached. Furthermore, we cache costs according to a keyframe strategy to bind the complexity in the case of long video sequences and ensure enough motion parallax. Following [39], a new incoming frame is promoted to keyframe if its relative translation and rotation are greater than \(t_{max}\) and \(R_{max}\) respectively. Then, a memory bank collects disparity \(d^3\) of the last \(N_{key}\) keyframes. Since each cached map represents the scene geometry in the past, to use it in the current computation we need to update the disparity values and their coordinates in the current image. Inspired by SLAM literature [29], we leverage a Local Map strategy to this aim. In detail, given the camera model \(\pi: \mathbb{R}^3 \rightarrow \mathbb{R}^2\), a 3D point \(P(X, Y, Z)\) can be projected to a 2D pixel \(p(u, v)\):

\[
\pi(P) = \left( \frac{f_x X}{Z} + c_x, \frac{f_y Y}{Z} + c_y \right),
\]

where \((f_x, f_y, c_x, c_y)\) are the known camera intrinsics. Similarly, a pixel \(p\) can be back-projected to a 3D point \(P\):

\[
\pi^{-1}(p, d) = \frac{b \cdot f_x}{d} \left( \frac{u - c_x}{f_x}, \frac{v - c_y}{f_y}, 1 \right)^\top,
\]

where \(b\) is the baseline between the left and the right cameras and \(d\) the disparity. With the relative extrinsic \(T_{j \rightarrow t} \in \mathbb{S}E(3)\) from keyframe \(j\) to the current frame \(t\), we can update every disparity value \(d_j\) of keyframe \(j\) as:

\[
d^{proj}_j = b \cdot f_x \cdot \frac{P_j(X_j, Y_j, Z_j) - T_{j \rightarrow t}^{-1} \cdot \pi^{-1}(p, d_j)}{Z_j},
\]

where \(d^{proj}_j\) is the updated disparity value. At this point, we obtain the coordinates of \(d^{proj}_j\) in \(t\) through forward warping.
To preserve end-to-end requirement, we use the differentiable Softmax Splatting \( \hat{d}_{\text{proj}} \) [30]:
\[
\hat{d}_{\text{proj}} = \sigma(d_{\text{proj}} - \pi(P_j) - p)
\]  

(10)

Finally, the Local Map is defined as follows:
\[
\mathcal{D}^2 = \mathcal{D}^2, \hat{d}_{\text{proj}}^2, d_{\text{proj}}^2, \ldots, \hat{d}_{\text{proj}}^{N_{\text{keys}}}
\]  

(11)

It is worth noticing that, in single-pair mode, the Local Map only contains candidates from the current pair. Moreover, it boosts candidate selection only for \( s = 2 \), since this stage represents the best trade-off between accuracy and speed. In fact, in \( s = 3 \) candidates are looked over the full search range at the lowest resolution, while in \( s = 3 \) a larger cost volume involves a much more expensive computation.

**Temporal shift and past costs.** Past semantic and matching scores are crucial for temporal stereo processing. Although Local Map effectively proposes previous depth cues about the scene, it lacks in providing the context behind these guesses. To address the problem, we introduce Temporal Shift and Past Costs modules. Temporal Shift enriches current feature maps with those computed in the past. Specifically, we adopt the TSM [23] strategy to facilitate the feature exchange among neighboring frames because it does not introduce additional computation or parameters: past features are shifted along the time dimension and then merged with current ones as shown in Fig. 2. Doing so, TSM provides spatio-temporal capabilities to our backbone not originally designed for temporal modeling. In practice, every feature map from the backbone is cached and used by Temporal Shift in the next prediction. Notably, as TSM does not rely on pose, we can still perform stereo matching in temporal mode and benefit from past information when pose is not available as shown in Tab. II.

Similarly, Past Costs module adds past matching scores to current cost volumes. Given \( C_1^f \) volume computed at time \( t-1 \), first we update the value of its candidates to \( t \) according to Eq. (9). Then, cost values and candidates are forward warped to \( t \) using Eq. (10). Finally, the warped cost volume is downscaled by a factor of 2 and 4 and concatenated with current \( C_2^\text{agg} \) and \( C_1^\text{agg} \) respectively, to be further aggregated by the Statistical Fusion module. To preserve computation, we only warp top- \( K \) candidates and their costs to \( t \). In single-pair mode, both Temporal Shift and Past Costs are the identity function. This strategy allows us to run the model trained in temporal mode also in single-pair mode with a limited drop in accuracy, e.g., at bootstrap.

**IV. EXPERIMENTS**

This section describes the experimental setups used to evaluate on popular datasets, including: SceneFlow [27], TartanAir [46], KITTI 2012 [11] and KITTI 2015 [28]. As standard practice in this field [57], [55], we compute the end-point-error (EPE) and the percentage of points with a disparity error > 3 pixels (3PE, > 5 for 5PE) as error metrics in non-occluded (NOC), occluded (OCC) and both (ALL) the regions. Moreover, following the literature [28], we measure the D1 error on KITTI instead of the 3PE in background (BG), foreground (FG), and both (ALL) areas. We do so by computing the percentage of points with error > 3 pixels and > 5% than the ground truth. Runtime reported in Tab I, II, VI is measured in the corresponding image resolution on each dataset, on a single NVIDIA 3090 GPU. The supplementary material outlines more details about implementation, dataset description, and training setups.

**A. Ablation Study**

**Single-pair mode.** We leverage SceneFlow [27] to assess the impact of main components of TemporalStereo in single-pair mode. Tab. I reports this ablation study, witnessing how each module helps to improve the results consistently. In particular, the multi-level cost computation with group-wise convolutions (B) is effective in enriching the cost volume of each stage, and the baseline model (A) also benefits from the further aggregation of Statistical Fusion (C). Nonetheless, the adaptive shifting strategy (E) provides without any doubt the main boost in performance. To be noticed, without the enriched context of each sparse candidate, the adaptive shifting strategy shows limitations on disparity correcting and its performance gain (D) decreases a lot. Furthermore, results shown in (F, G, H) illustrates our TemporalStereo is able to predict accurate disparity with as few as 5 candidates. With the ability to shift the candidates towards a better solution, our model can search in a quite large space (e.g., \( \beta = 2, 4, 8 \) as reported in H, I, J) to get the best result when \( \beta = 4 \). 3D convolutions are the common operations to aggregate the cost in recent methods [3], [19], [57]. Our baseline model (A) benefits from the 3D convolutions (K) when compared to depth-wise [32] 3D convolutions (L), but the runtime increases a lot. In contrast, the full model (E) gives much higher improvement (N) with negligible runtime increase (4.8ms), and the time-consuming 3D convolutions are not necessary (M) for accuracy improvement.

**Temporal mode.** Before presenting the results achieved in temporal mode, we illustrate the protocol adopted at training and test time on TartanAir dataset [46]. Based on a temporal

| Candidates | Disparity Range | EPE | Runtime (ms) |
|------------|----------------|-----|--------------|
| (A)        |                 |     |              |
| (B)        |                 |     |              |
| (C)        |                 |     |              |
| (D)        |                 |     |              |

**TABLE II: Temporal mode ablation.** We evaluate the temporal components on TartanAir, with \( W_{tr} = W_{test} = 4 \).
window containing $W$ frames, initial $f = \{1, 2, ..., W - 1\}$ frames are processed by the network sequentially without computing error metrics and blocking the gradients. Specifically, each outcome is cached and used in the next frame prediction. When $f = W$, we compute errors (and backpropagation at training time). Tab. II reports the ablation conducted on TartanAir [46], with $W = 4$ both at train and test time, aimed at evaluating the importance of each temporal module. Specifically, we train the model in the single-pair mode for $20$ epochs (i.e., the model learns how to solve stereo matching task), then we enable temporal components for $20$ more epochs (i.e., the model now focuses on how to use past information). We can notice how the baseline (A), i.e., the model trained in single-pair mode for $40$ epochs, could be improved using Temporal Shift to fuse past features with current ones (B). However, the benefit due to Temporal Shift is much lower than the gain provided by Local Map, which largely improves the performance (C). Finally, including cached past costs as well (D) provides an additional boost in accuracy.

**Impact of temporal window.** Tab. III reports the results achieved by TemporalStereo using different values of $W$. In particular, we could have two different values for $W$, that are $W_{tr}$ and $W_{test}$ for train and test respectively. In addition to the baseline $W_{tr} = 1$ and the temporal $W_{tr} = 4$ models, we also train a $W_{tr} = 8$ model following the same configuration as for $W_{tr} = 4$. When compared with existing models such as PSMNet [6] and CoEx [3], our baseline outperforms them by a large margin in EPE metric. As for temporal mode, in general, the more frames joining the training or testing phases, the better result we can get in all regions. Notably, $W_{test} = 4$ and $W_{test} = 8$ are always beneficial in OCC, witnessing that TemporalStereo can effectively exploit more frames to deal with such difficult areas. Moreover, when $W_{test} > 1$, the results consistently overcome baseline with single stereo pair. It indicates our network can benefit from past information with only a few frames (e.g., 4, 8) after the network startup. Finally, we can notice how temporal models tested with $W_{test} = 1$ obtain results closer to the baseline. This outcome implies that, in practical video applications, TemporalStereo can be trained once in temporal mode and used in single-pair way for the first inference (yet providing a good initial estimate) and in the temporal mode for all the others. Fig. 4 visualises the benefits of temporal model ($W_{tr} = 8$, $W_{test} = 8$) to alleviate occlusion errors. The temporal mode (2) largely outperforms the single-pair one (1). Furthermore, we highlight how TemporalStereo is also robust against inaccurate poses: although in (3) the camera pose is set to an identity matrix and only Temporal Shift module could help, it still outperforms (1). Eventually, the proposed temporal cues can be plugged into recent efficient methods (e.g., CoEx [3] and StereoNet [19]) for further improvement. Although CoEX with a very different architecture compared to ours, its EPE still decreases from 0.714 to 0.610, with near 14% improvement. Nonetheless, our TemporalStereo utilizes past cues more effectively.

**Pose Analysis.** To assess the impact of pose input, we evaluate our model with setting $W_{tr} = 8$ and $W_{test} = 8$ on 4 scenes (2 indoor and 2 outdoor scenes) of TartanAir dataset [46] with hard motion patterns. The overall results of several pose inputs are listed in Tab IV. Specifically, the noise of rotation follows $N(0, \sigma_R)$ in degree($^\circ$) and noise of translation follows $N(0, \sigma_t)$ in meter($m$). As reported, 1) poses from ground truth (GT) or estimated by DROID-SLAM [43] yield almost the same EPE and 3PE metrics. It means, when the ground truth pose is not available, an actual SLAM system like DROID-SLAM could be an alternative scheme. 2) Pose with small rotation error (e.g., $\sigma_R < 1^\circ$) and translation error (e.g., $\sigma_t <= 0.05 m$) gets very close results to the one with ground truth pose, and it always surpasses the results by single-pair mode. Even when the pose error comes to the maximum level of the dataset (i.e., $\sigma_R = 10^\circ$, $\sigma_t = 0.5 m$), our model does not crash down and provides impressive predictions, proving the robustness of our model to inaccurate pose. 3) For outdoor scenes, since the view change between frames is much smaller than indoor scenes, the accuracy drops negligibly even when identity transformation is applied.

### Table III: Impact of different frames in temporal mode.

Models are tested with $W_{test} = 1, 4$ and $8$.

| Method                | $W_{test}$ | EPE | 3PE | EPE | 3PE |
|-----------------------|------------|-----|-----|-----|-----|
|                       | ALL | OCC | NOC | ALL | OCC | NOC |
| PSMNet [6]            | 1   | 0.866 | 2.654 | 0.558 | 5.493 | 2.486 |
| StereoNet [19]        | 1   | 0.886 | 2.647 | 0.576 | 5.515 | 2.466 |
| CoEx [3]              | 1   | 0.714 | 2.074 | 0.463 | 3.583 | 1.957 |
| Ours (single-pair)    | 1   | 0.647 | 1.899 | 0.428 | 3.996 | 1.211 |
| Ours (temporal)       | 4   | 0.610 | 1.637 | 0.413 | 3.733 | 1.260 |
| Ours (temporal)       | 8   | 0.607 | 1.634 | 0.409 | 3.711 | 1.259 |
| Ours (temporal)       | 8   | 0.601 | 1.615 | 0.405 | 3.716 | 1.254 |

![Fig. 4: Benefits of temporal mode.](image)

To conclude, we run TemporalStereo on KITTI 2012 and KITTI 2015 test data and submit to the online leaderboard.

### Pretraining.

As KITTI is very challenging due the lack of a big training set, pretraining on SceneFlow dataset [27] is a common training schedule for deep learning-based stereo methods [6], [7]. Recent methods [36], [51] also introduce extra data, e.g., HR-VS [51], to augment KITTI during training. Following the knowledge distillation strategy proposed in AANet+ [50], we augment the KITTI dataset by leveraging the prediction results from pretrained LEAStereo [7] to
TABLE IV: Impact of different pose input in temporal mode. “Single-pair” denotes our model is in single-pair mode (no pose needed); “Identity” means identity matrix; “DROID-SLAM” means pose estimated by DROID-SLAM (ORB-SLAM3 fails on those 4 scenes); and “GT+Noise” is the ground truth pose with manually added Gaussian noise.

| Pretrain | SF       | SF+Pseudo | TartanAir+Pseudo |
|----------|----------|-----------|-----------------|
| D1-BG    | 1.79     | 2.17      | 1.89            |
| D1-FG    | 3.82     | 2.96      | 2.85            |
| D1-ALL   | 2.13     | 2.30      | 2.05            |

TABLE V: Impact of pretraining. We study the importance of pretraining by evaluating the D1 metric on KITTI 2015 test dataset. Results by both CoEx and Ours in single-pair and temporal mode (†) are reported accordingly. ‘SF’ and ‘TartanAir’ denotes SceneFlow and TartanAir datasets respectively. ‘+Pseudo’ means further training on KITTI raw sequences with our generated pseudo label.

TABLE VI: Comparison of state-of-the-art methods – slow and fast. We report results of state-of-the-art methods on SceneFlow and KITTI, fast denotes models allowing real-time inference. ‡ denotes 3D scene flow-based methods.

KITTI raw sequences [9]. In single-pair mode, our TemporalStereo achieves results already on par with state-of-the-art on all datasets. When switching to the temporal mode, our network surpasses all fast stereo architectures by a large margin. In particular, our result on D1-FG is even better than the one achieved by slow models [7], [55] running in hundreds of milliseconds. It is worth mentioning that TemporalStereo in temporal mode, benefiting from past semantic and geometric information, achieves better results compared to single-pair mode on D1-FG metric (the D1 error on foreground areas, i.e., moving cars), which proves the robustness of our model to dynamic objects as well. The same advantage is also evident on 3PE and 5PE in reflective regions on KITTI 2012. Finally, we also compare with 3D scene flow based state-of-the-art methods [1], [35], [17], which project with dense 3D motion field and output disparity. In contrast, simply using camera pose, our TemporalStereo is the obvious winner in both accuracy and efficiency on KITTI 2015.

V. CONCLUSIONS AND LIMITATIONS

We presented TemporalStereo, a novel network devoted to fast stereo matching. The enhanced coarse-to-fine design and sparse cost volumes allow for fast inference and high performance. Moreover, TemporalStereo can exploit past information to ameliorate predictions, especially in occluded regions. The same model, trained once, can handle either single or multiple stereo pairs effectively. Considering the requirement of camera poses as its main limitation, empowering our system with pose estimation will be our future research direction.

Acknowledgment. We sincerely thank the scholarship supported by China Scholarship Council (CSC).

REFERENCES

[1] Filippo Alcotti, Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning end-to-end scene flow by distilling single tasks knowledge. In AAAI, 2020.
[2] Abhishek Badki, Alejandro Troccoli, Kihwan Kim, Jan Kautz, Pradeep Sen, and Orazio Gallo. Bld3: Stereo depth estimation via binary classifications. In CVPR, 2020.
[3] Antyanta Bangunharana, Jae Won Cho, Seokju Lee, In So Kweon, Kyung-Soo Kim, and Soohyun Kim. Correlate-and-excite: Real-time stereo matching via guided cost volume excitation. In IROS, 2021.
[4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM TOG*, 28(3), 2009.

[5] Carlini, Richard Elvira, Juan A. Gómez, José M. M. Montiel, and Juan D. Tardó. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM, 2020.

[6] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, 2018.

[7] Xuelian Cheng, Yiran Zhong, Muthuraj Haranadi, Yuchao Dai, Xiaojun Chang, Tom Drummond, Hongdong Li, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *NeurIPS*, 2020.

[8] Shihv Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun. Deeppmrner: Learning efficient stereo matching via differentiable patchmatch. In *ICCV*, 2019.

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 2017.

[10] Divyang Garg, Yan Wang, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Wasserstein distances for stereo disparity estimation. *NeurIPS*, 2020.

[11] Andreas Geiger, PhilipLenz,and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*. IEEE, 2012.

[12] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozech Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *CVPR*. June 2020.

[13] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiangang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *CVPR*, 2019.

[14] Marc Hirschmenger. Stereo processing by semiglobal matching and mutual information. *IEEE TPAMI*, 30(2), 2007.

[15] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *CVPR*, 2019.

[16] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for video classification. In *CVPR*, 2018.

[17] Huaizu Jiang, Deqing Sun, Varun Jampani, Zhaoyang Lv, Erik Learned-Miller, and Jan Kautz. Sense: A shared encoder network for scene-flow estimation. In *ICCV*. October 2019.

[18] Alex Kendall, Holly Martirosyan, Saumit Dutagupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, 2017.

[19] Sanch Khamsin, Kean Fannello, Christoph Rhemann, Adarsh Kowdle, Julian Valentini, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *ECCV*, 2018.

[20] Hsueh-Ying Lai, Yi-Hsuan Tsai, and Wei-Chen Chen. Bridging stereo matching and optical flow via spatiotemporal correspondence. In *CVPR*, June 2019.

[21] Jiankun Li, Peten Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jianguo Liu, Haoqiang Fan, and Shuaicheng Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation. In *CVPR*, 2022.

[22] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhui Guo, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang. Learning for disparity estimation through feature constancy. In *CVPR*, 2018.

[23] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *ICCV*, 2019.

[24] Pengpeng Liu, Irwin King, Michael R Lyu, and Jia Xu. Flow2stereo: Effective self-supervised learning of optical flow and stereo matching. In *CVPR*, 2020.

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *CVPR*, 2016.

[26] Yamin Mao, Zhihua Liu, Weiming Li, Yuchao Dai, Qiang Wang, Yun-Tae Kim, and Hong-Seok Lee. Lissnet: Uncertainty adaptive sampling network for deep stereo matching. In *ICCV*, 2021.

[27] Nikolaus Mayer, Eddy Ilg, Philip Hauser, 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 *CVPR*, 2016.

[28] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.

[29] Raul Mur-Artal and Juan D. Tardós. ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5), 2017.

[30] Simon Niklaus and Feng Liu. Softmap splatting for video frame interpolation. In *CVPR*, 2020.

[31] Matteo Poggio, Fabio Tosi, Konstantinos Batson, Philippos Mordohai, and Stefano Mattoccia. On the synergies between machine learning and binocular stereo for dense scene flow estimation from images: A survey. *IEEE TPAMI*, 44(9):5314–5334, 2022.

[32] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[33] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3), 2002.

[34] Korbinian Schmid, Teodor Tomic, Felix Rußko, Heiko Hirschmüller, and Michael Suppa. Stereo vision based indoor/outdoor navigation for flying robots. In *IROS*, IEEE, 2013.

[35] Rene Schuster, Christian Unger, and Didier Stricker. A deep temporal fusion framework for scene flow using a learnable motion model and occlusions. In *WACV*, 2021.

[36] Zheyun Shen, Yuchao Dai, and Zhibo Rao. Cnet: Cascade and fused cost volume for robust stereo matching. In *CVPR*, June 2021.

[37] Sayan Sivaraman and Mohan M Trivedi. A review of recent developments in vision-based vehicle detection. In *IV*, IEEE, 2013.

[38] Xiao Song, Xu Zhao, Hanwen Hu, and Liangjiang Fang. EdgeStereo: A context integrated residual pyramid network for stereo matching. In *ACCV*, Springer, 2018.

[39] Jiamei Sun, Yiming Xie, Linghao Chen, Xaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *CVPR*, 2021.

[40] Vladimir Tankovich, Christian Hane, Yingda Zhong, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. HNet: Hierarchical iterative tile refinement network for real-time stereo matching. In *CVPR*, June 2021.

[41] Zachary Teed and Jia Deng. DeepV: Video to depth with differentiable structure from motion. *ICLR*, 2020.

[42] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *CVPR*, 2020.

[43] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *NeurIPS*, 2021.

[44] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *CVPR*, 2021.

[45] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *CVPR*, 2021.

[46] Wenshan Wang, Delong Zhu, Xiaowei Wang, Yuheng Qu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *IROS*, 2020.

[47] Yimin Wang, Zhang Lai, Gao Huang, Brian H Wang, Lauren Van Der Maaten, Mark Campbell, and Kilian Q Weinberger. Anytime stereo image depth estimation on mobile devices. In *ICRA*, IEEE, 2019.

[48] Yang Wang, Peng Wang, Zhenheng Yang, Chenxu Lao, Yi Yang, and Wei Xu. Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. In *CVPR*, 2019.

[49] Gangwei Xu, Junda Cheng, Peng Guo, and Xin Yang. Attention concatenation volume for accurate and efficient stereo matching. In *CVPR*, 2022.

[50] Jiaofei Xu and Juoyang Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *CVPR*, 2020.

[51] Gengshan Yang, Joshua Manela, Michael Hapgood, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *CVPR*, June 2019.

[52] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhiqiong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *ECCV*, 2018.

[53] Qingxiong Yang. A non-local cost aggregation method for stereo matching. In *CVPR*. IEEE, 2012.

[54] Nadia Zenati and Noureddine Zerhouni. Dense stereo matching with application to augmented reality. In *MobileNetv2*, June 2021.

[55] Feihu Zhang, Oliver J. Woodford, Victor Adrian Prisacariu, and Justin北. On the synergies between machine learning and stereo vision. In *CVPR*, 2018.

[56] Feihu Zhang, Oliver J. Woodford, Victor Adrian Prisacariu, and Justin North. On the synergies between machine learning and stereo vision. In *CVPR*, 2018.

[57] Youmin Zhang, Yimin Chen, Xiao Bai, Suihanjin Yu, Kun Yu, Zhiwei Li, and Kuiyuan Yang. Adaptive unimodal cost volume filtering for dense stereo matching. In *ICCV*, 2019.

[58] Youmin Zhang, Zhiping Huang, Yuwen Zhou, and Bin Liu. Dynamic cost volume filtering for robust stereo matching. In *CVPR*, 2019.