Active Stereo Without Pattern Projector

Luca Bartolomei∗†, Matteo Poggi†, Fabio Tosi†, Andrea Conti†, Stefano Mattoccia∗†

∗Advanced Research Center on Electronic System (ARCES)  †Department of Computer Science and Engineering (DISI)
University of Bologna, Italy
{lucabartolomei5, m.poggi, fabio.tosi5, andrea.conti35, stefano.mattoccia}@unibo.it
https://vppstereo.github.io/

Figure 1: Virtual Pattern Projection for deep stereo. Either in challenging outdoor (top) or indoor (bottom) environments (a), a stereo network such as PSMNet [8] often struggles (b). By projecting a virtual pattern on images (c), the very same network dramatically improves its accuracy (d). Further training the model to deal with the augmented images (e) improves the results even more.

Abstract

This paper proposes a novel framework integrating the principles of active stereo in standard passive camera systems without a physical pattern projector. We virtually project a pattern over the left and right images according to the sparse measurements obtained from a depth sensor. Any such devices can be seamlessly plugged into our framework, allowing for the deployment of a virtual active stereo setup in any possible environment, overcoming the limitation of pattern projectors, such as limited working range or environmental conditions. Experiments on indoor/outdoor datasets, featuring both long and close-range, support the seamless effectiveness of our approach, boosting the accuracy of both stereo algorithms and deep networks.

1. Introduction

Depth perception is crucial in several computer vision tasks, including autonomous driving, 3D reconstruction, robotics, and augmented reality. Inferring depth from standard cameras, according to different setups and strategies, is one of the most widely deployed techniques due to its low cost and potentially unbounded image resolution. At the core of these approaches, using multiple cameras or a moving one, there is the problem of determining visual correspondence. However, matching points across frames is inherently ambiguous in the presence of textureless regions, repetitive patterns, and non-Lambertian materials. This task is even more challenging when performed densely for each pixel in the input images. Although deep learning has achieved excellent results, as witnessed by the recent literature [53], it is also prone to the well-known domain shift issue absent in conventional, less accurate hand-crafted methods. Specifically, since learning-based methods rely on training data, they suffer severe drops in accuracy when facing different data distributions [52]. A different approach to depth perception relies on active sensing technologies, such as LiDAR (Light Detection and Ranging), ToF (Time of Flight), and Radar (Radio Detection and Ranging). However, each technology has limitations. LiDAR technology is reliable but features a density much lower than the resolution of modern cameras, making it extremely expensive as density increases. ToF suffers from the same limitation and is also unreliable under sunlight and at longer distances. Radar allows for a more extended depth range sensing, but it is sparser, noisier, and with a narrower vertical field of
view [39]. Finally, active systems that estimate depth from images also exist, relying on structured [23] or unstructured [31] pattern projection results more accurate than passive imaging techniques and at higher resolution with respect to the aforementioned devices. However, these systems are bounded by the need for the projected pattern to be clearly visible in images, and thus cannot work beyond very close distances (i.e., a few meters), are unsuited for outdoor use with sunlight, and the presence of multiple projectors might make them interfere.

Due to their complementary strengths and limitations, setups made of active and passive technologies are widespread for several application fields, ranging from autonomous driving, where almost all prototypes have heterogeneous sensor suites, to augmented reality with smartphones and tablets equipped with cameras and active depth sensors. Consequently, different solutions exist in the literature to exploit the synergy between active and passive depth sensing [51, 13, 73]. The common key trait of most of these sensor-fusion methods consists in modifying the internal behavior of the camera-based stereo matcher or concatenating the sparse points with the color images. In contrast, this paper proposes a novel paradigm to leverage the synergy between active and passive sensing. It works by coherently hallucinating the vanilla stereo pair acquired by a standard camera simplifying the visual correspondence task performed by any stereo network algorithm as if a virtual pattern projector were present in the scene. Such virtual coherent pattern projection is feasible by exploiting the stereo geometry and a registered active depth sensor providing sparse yet accurate measurements, like in [51, 13, 73]. Our proposal shares the same motivations of active methods based on unstructured pattern projection. However, unlike these strategies, it does not rely on a specific physical pattern projector with all the limitations outlined previously. Instead, by selecting the depth sensor that is better suited for the specific scenario, our approach can work in any environment and is agnostic to moving objects and camera ego-motion, as shown in Fig. 1. Experimental results on standard stereo datasets support the following claims:

- Even with meager amounts of sparse depth seeds (e.g., 1% of the whole image), our approach outperforms by a large margin state-of-the-art sensor fusion methods based on handcrafted algorithms and deep networks.
- When dealing with deep networks trained on synthetic data, it dramatically improves accuracy and shows a compelling ability to tackle domain shift issues, even without additional training or fine-tuning.
- By neglecting a physical pattern projector, our solution works under sunlight, both indoors and outdoors, at long and close ranges with no additional processing cost for the original stereo matcher.

We believe that our proposal, dubbed Virtual Pattern Projection (VPP), has the potential to become a standard component for depth perception and pave the way to exciting future developments in the field.

2. Related Work

**Stereo Matching.** Traditional stereo algorithms [85, 72, 81, 80, 37, 65, 33, 27, 5], thoroughly investigated in [58], rely on handcrafted features and priors to compute dense disparity maps from stereo pairs. Deep learning has recently revolutionized stereo matching, providing significant improvements over conventional techniques on standard benchmarks [86]. Specifically, end-to-end stereo networks have become the most popular and effective solution for disparity estimation. These networks can be categorized into two main families: 2D and 3D architectures, with the former adopting an encoder-decoder design [46, 48, 38, 55, 62, 79, 82, 66], inspired by the U-Net model [54], while the latter build a feature cost volume from features extracted on the image pair and estimate the disparity map using 3D convolutions [30, 8, 32, 87, 12, 14, 19, 78, 74, 25, 61], at the cost of a much higher memory requirement and runtime. A complete review of such works can be found in [53]. More recent works [41, 35], instead, propose novel deep stereo networks that exploit the iterative refinement paradigm in the state-of-the-art optical flow network RAFT [67], or rely on Vision Transformers [36, 24] to capture long-range contextual information. Despite their superior performance, deep learning-based stereo methods require amounts of annotated data for training and yet suffer from limited generalization capabilities to unseen data [88, 6, 1, 90, 42, 16, 76, 71]. To alleviate this requirement, self-supervised techniques have been proposed to train deep stereo models without ground-truth annotations by relying on photometric losses on stereo pairs or videos [93, 70, 69, 34, 75, 15], traditional algorithms and confidence measures [68, 2]. Others propose self-supervised continual adaptation of stereo networks using images captured during deployment [70, 52].

**Active Stereo Matching.** Active stereo leverages a pattern projector, typically working in the IR spectrum, to ease the correspondence problem, particularly in low-textured and repetitive areas. These systems [9] rely on structured [23] or unstructured [31] light. A subclass of structured light systems – coded light – relies on spatial [26] or temporal coded patterns [84], with the latter struggling in the presence of moving objects at standard frame rate. Regardless of the technology, pattern projection is limited to shorter distances and is unsuitable for outdoor use with sunlight. Additionally, devices based on structured light patterns require carefully designed projectors and suffer from thermal drifts [21] among other issues [56]. Nonetheless, active stereo is a vivid research field [43, 20].
Figure 2: Overview of the proposed Virtual Pattern Projection framework. The envisioned setup on the left relies on a standard passive stereo camera and an active depth sensor. On the right, the step needed to obtain the coherently hallucinated stereo pair from the vanilla input images and sparse depth points through Virtual Pattern Projection (VPP). The patterned stereo pair in output can be processed by any stereo matcher, either handcrafted or learning-based.

Image-Guided Methods. Another approach to estimating depth involves integrating sparse depth measurements from active sensors with RGB information. Specifically, two main trends are typically adopted in the literature.

Depth Completion. This approach aims to estimate dense depth maps from incomplete or noisy depth data or from a single RGB image combined with sparse depth measurements, which can be obtained from active sensors such as LiDAR or structured light. A wide range of methods have been proposed for this task, including traditional methods based on interpolation and optimization [7, 60, 44], as well as deep learning-based approaches [45, 11, 49, 28, 10, 40, 91]. However, these approaches typically suffer in estimating depth in regions with missing external depth values [18].

Guided Stereo Matching. In addition to advances in stereo matching, recent research has also explored the integration of stereo with active sensors. In Badino et al. [3], the authors propose to directly integrate LiDAR data into the stereo algorithm using dynamic programming. Gandhi et al. [22], instead, propose a method to fuse time-of-flight (ToF) camera and stereo pairs using an efficient seed-growing algorithm. More recent works, instead, exploit depth measurements, either by concatenating them as input of CNN-based architectures [13, 2, 50, 89, 73] or by using them to guide the cost aggregation of existing cost volumes [51, 29, 92, 73]. Our approach augments the given stereo pair to enhance RGB images, providing more discriminative information to the network and making it easier to solve the correspondence problem, unlike other methods that only concatenate sparse depths or guide cost aggregation.

3. Virtual Pattern Projection (VPP)

This section describes our approach to ease visual correspondence across images acquired by a passive stereo camera by means of the coherent virtual projection of patterns leveraging the availability of reliable sparse depth points of the same scene. The motivation for enriching the visual image content of the original scene is the same as methods relying on a physical projector: it aims at increasing local distinctiveness to make the visual correspondence task more robust. However, in contrast to previous methods, our strategy follows an entirely different path discarding the need for a physical projector with all its mentioned limitations.

3.1. Virtual Projection Principle

Our proposal is based on the observation that given a rectified stereo rig, and availability of a set of sparse depth points registered with the reference image, each of these points implicitly allows us to determine the corresponding pixels in the stereo pair. Hence, by knowing this correspondence, we can augment the visual appearance of both pixels in the two images to make them as similar as possible and as distinctive as possible from their neighbors, as if an ideal smart virtual projector were sending its signal onto the sensed scene. Fig. 2 outlines our basic virtual projection principle, showing on the left the envisioned setup consisting of a calibrated stereo camera and a sensor performing sparse yet accurate depth estimation, registered with the stereo reference camera through an initial calibration. The vanilla stereo pair acquired by the passive cameras and the input depth map are shown at the top. Since the stereo rig and the active sensor are calibrated, the depth \( z(x, y) \) of each sparse point allows us to locate corresponding points.
Figure 3: Virtual patterns. From top to bottom: (i) indistinctive, (ii) randomly generated, (iii) distinctive, (iv) randomly generated patch-based uniform, (v) distinctive patch-based uniform, (vi) randomly generated patch-based, and (vii) distinctive patch-based patterns. For (iii), (v), and (vii) we report the search area of length $L$, set to 64 in our experiments, and height as depicted for $3 \times 3$ virtual patterns.

$I_L(x, y)$ and $I_R(x', y)$ in the two input images. For this purpose, we transform $z(x, y)$ into a disparity $d(x, y)$ by knowing [64] the focal length $f$ and the baseline $b$ of the stereo camera with $d(x, y) = \frac{b}{z(x, y)}$. Such a disparity value represents the offset needed to obtain the location along the same epipolar line of the corresponding point $I_R(x', y)$ in the target image with $x' = x - d(x, y)$. Fig. 2 shows how the two corresponding points $A$ and $B$ can be determined by knowing the depth of $A$ in the input sparse depth map, and then we can augment them to achieve our goal.

### 3.2. Virtual Patterns

To properly hallucinate images and ease matching, corresponding points should be as similar as possible. Accordingly, we propose two augmenting strategies: random patterning and histogram-based pattern. Both operate on corresponding points $(x, y)$ and $(x', y)$ located along the epipolar line with the same pattern $A(x, x', y)$ in the two images:

$$I_L(x, y) \leftarrow A(x, x', y)$$
$$I_R(x', y) \leftarrow A(x, x', y)$$  \hspace{1cm} (1)

Their difference consists in how the operator $A(x, x', y)$ generates the virtual pattern superimposed on the input images. Fig. 3 outlines the several possible patterns that we will discuss in the remainder. Furthermore, as $x'$ is unlikely to be an integer, $A(x, x', y)$ will apply to both $I_R([x'], y)$ and $I_R([x'], y)$ by means of a weighted splatting on the two.

$$I_R([x'], y) \leftarrow \beta I_R([x'], y) + (1 - \beta) A(x, x', y)$$
$$I_R([x'], y) \leftarrow (1 - \beta) I_R([x'], y) + \beta A(x, x', y)$$  \hspace{1cm} (2)

where $\beta = x' - [x']$ is the splatting weight.

3.2.1 Random Patterning

To ease visual correspondence, the pattern should both increase similarity across images, as well as distinctiveness [83] along the epipolar line. As such, a pattern as the one shown in Fig. 3 (i) would be suboptimal. To address this issue, a method uses an operator $A(x, x', y)$ that samples a random value from a uniform distribution

$$A(x, x', y) \sim U(0, 255)$$  \hspace{1cm} (3)

Fig. 3 (ii) depicts a possible outcome of this approach whose extension to color images, as for any method discussed, is straightforward by applying the same strategy on each color channel. This operator is fast and introduces distinctiveness to some degree, although not entirely. On the contrary, an operator taking into account the image content itself could enforce stronger distinctiveness [83].

3.2.2 Histogram-based Patterning

We argue that the patterns superimposed onto the vanilla image should stand out from the background and be unambiguous (at least) within nearby pixels along the same horizontal scanline, as depicted in Fig. 3 (iii).

To achieve this goal, we employ a histogram-based operator $A(x, x', y)$, to select the pattern by analyzing the scanline content in the two hallucinated images. For $(x, y)$ in the reference image, we consider two windows of height 3 and length $L$ centered on it and on $(x', y)$ in the target image. Then, the histograms computed over the two windows are summed up and the operator $A(x, x', y)$ picks the color maximizing the distance from any other color in the histogram $H(i)$, with $h(i)$ returning the minimum distance from a filled bin in the sum histogram $H$.
3.3. Advanced Virtual Patterns

We now extend the strategies described so far, taking into account locality, the original image content, and occlusions.

3.3.1 Locality

The pointwise patterning strategy can be applied to larger areas to enhance the visual appearance further. To this purpose, we can extend previous methods to patches (e.g., $3 \times 3$ or $5 \times 5$), implicitly assuming the same disparity within these small regions. Fig. 3, in (iv) and (v), shows the outcome of the operator $A(x, x', y)$ that selects a uniform color within the patches according to, respectively, random sampling or a selection based on histograms in a larger region $L$. Further patterns can be generated by (vi) random sampling or (vii) histogram-base selection being performed for every pixel in the patch independently.

3.3.2 Alpha-Blending

Although a virtual pattern eases the match for traditional algorithms [27], it might hinder a deep stereo model not used to deal with it. Thus, we combine the original image content with the virtual pattern through alpha-blending [64] as follows, being $\alpha$ a hyperparameter:

$$I_L(x, y) \leftarrow (1 - \alpha)I_L(x, y) + \alpha A(x, x', y)$$
$$I_R(x', y) \leftarrow (1 - \alpha)I_R(x', y) + \alpha A(x, x', y)$$

3.3.3 Occlusions

Since occluded regions inevitably exist in a stereo setup, even assuming a depth sensor perfectly aligned with the reference camera (although this is not always the case [17]), we might not be able to project the pattern consistently on the two views, as depicted in Fig. 4 I). Accordingly, it is crucial to detect points hitting occluded regions to avoid projecting the same pattern on both the occluded and the occluder pixels, respectively, on the reference and target images. Purposely, we devise a simple yet effective heuristic to classify sparse disparities warped onto the target image, according to the difference in disparity and spatial distance from other sparse points inferred by the sensor.

Specifically, we warp disparity $d$ for $(x, y)$ into an image-like grid $W$ at coordinates $(x', y)$. In case of collisions – i.e., multiple $d$ warped at the same location $(x', y)$ – the largest $d$ is kept. Then, each $(x_o, y_o)$ in $W$ is classified as occluded if the following inequality holds for at least one neighbor $W(x, y)$ within a $r_x \times r_y$ patch:

$$W(x, y) - W(x_o, y_o) - \lambda (\gamma |x - x_o| + (1 - \gamma) |y - y_o|) > t$$

with $\lambda, \gamma, r_x, r_y, t$ being hyper-parameters. Finally, the occluded points are warped back to obtain a mask $o$.

When a depth point is classified as occluded, we can neglect projection on both reference and target images (“NO” projection strategy). This avoids projecting the same pattern on the foreground (in the target image) and background (in the reference), which would increase ambiguity at occlusions – “BKGD” projection strategy. Nonetheless, we
follow a third strategy: we avoid projection and instead replace the original content in \((x, y)\) on the reference image with the content at \((x', y)\) in the target image. This does not alter the appearance of the correct match on foreground (rays originating from Q), yet stimulates the stereo matcher to establish a second correspondence with the same point \((x', y)\) in the target image, i.e., with pixel \((x, y)\) originating from P. This strategy will be referred to as “FGD” projection. Additionally, points in the left border of the reference image would project patterns outside the target. Although irrelevant for traditional algorithms, we still project there to avoid artifacts in the predictions by deep stereo networks. Fig. 5 shows an example of hallucinated pair.

4. Experimental Results

We describe our experimental setup, including implementation details, datasets, and results analysis.

4.1. Implementation and Experimental Settings

All virtual pattern variants depicted in Fig. 3 from (ii) to (vii) are implemented in Cython, sub-pixel disparities splatted on adjacent pixels in the right view. Among the hyper-parameters, we set \(\lambda = 2, \gamma = 0.4375, t = 1\) and \(r_x \times r_y = 9 \times 7\) for the occlusion detection heuristic, whose effectiveness is studied in the supplementary material. To assess the effectiveness of VPP, we run several experiments in comparison with existing approaches that combine sparse depth points with stereo algorithms and networks. In particular, we consider the Guided Stereo Matching framework [51], LidarStereoNet [13], and CCVNNorm [73] as main competitors.

Since the first two approaches are implemented over the PSMNet [8] architecture, we apply VPP to PSMNet for a direct and fair comparison. However, the original PSMNet weights used in [51] yielded poor generalization results; therefore, we retrain it following the original protocol [8], i.e., for 10 epochs on SceneFlow with a constant learning rate equal to 1e-3. For fairness, LidarStereoNet and CCVNNorm have been retrained using the same setting. Moreover, since guided stereo [51] represents our closest competitor – i.e., it can be applied without any architectural change to the stereo model – we implement it within a more modern network, RAFT-Stereo [41], and compare it against our VPP. To conclude, we extend this comparison with an implementation of the Semi-Global-Matching method [27], i.e., rSGM [63], as a representative of traditional stereo algorithms. We run it by setting the maximum disparity to 192 disparity \(P_1=11\), adaptive \(P_2\) [4] (with \(P_{2_{\text{max}}}=17\), \(P_{2_{\alpha}}=0.5\), \(P_{2_{\gamma}}=35\)), applying left-right check and a speckle filter to remove outliers, then filling holes with background interpolation [47].

4.2. Evaluation Datasets & Protocol

We run experiments on three indoor/outdoor datasets.

**Middlebury.** The Middlebury dataset [57] is a high-resolution stereo dataset featuring indoor scenes, captured under controlled lighting conditions with accurate ground-truth obtained using structured light. We evaluate our approach on three different splits: the Additional 13 scenes in Middlebury 2014 (Mid-A), the 15 scenes from Middlebury 2014 training set (Mid-14), and the 24 scenes from Middlebury 2021 (Mid-21). Results are evaluated at full resolution, with PSMNet models running at half and CCVNNorm at quarter (Mid-14) and one third (Mid-21) resolution.

**KITTI 2015 [47].** This real-world stereo dataset depicts autonomous driving scenarios, captured at a resolution of approximately 1280×384 pixels with sparse ground-truth depth maps collected using a LiDAR sensor. It provides 200 stereo pairs annotated with ground-truth, including in-

### Table 1: Ablation on main projection hyperparameters. Results on Midd-A. Networks trained on synthetic data.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Hyperparameters</th>
<th>Error Rate (%) &gt; 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\alpha), (\gamma)</td>
<td>RAFT-Stereo [41]</td>
</tr>
<tr>
<td>(i)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(ii)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(iii)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(iv)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(v)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(vi)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(vii)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1: Ablation on main projection hyperparameters. Results on Midd-A. Networks trained on synthetic data.
<table>
<thead>
<tr>
<th>Model</th>
<th>Depth Points</th>
<th>Error Rate (%)</th>
<th>avg (px)</th>
<th>Error Rate (%)</th>
<th>avg (px)</th>
<th>Error Rate (%)</th>
<th>avg (px)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train</td>
<td>Test</td>
<td>&gt; 1</td>
<td>&gt; 2</td>
<td>&gt; 3</td>
<td>&gt; 4</td>
<td>&gt; 1</td>
</tr>
<tr>
<td>rSGM [51]</td>
<td>✓ ✓</td>
<td>38.07 30.64 17.03 13.57</td>
<td>9.98</td>
<td>19.12 26.56 13.51 10.04</td>
<td>4.44</td>
<td>19.95 7.18 1.72 1.28</td>
<td>0.76</td>
</tr>
<tr>
<td>rSGM-gd [51]</td>
<td>✓ ✓</td>
<td>57.98 20.37 10.86 9.49</td>
<td>7.81</td>
<td>52.21 15.08 6.63 6.51</td>
<td>3.40</td>
<td>9.66 0.62 0.46 0.41</td>
<td>0.58</td>
</tr>
<tr>
<td>rSGM-pp</td>
<td>✓ ✓</td>
<td>45.22 33.11 24.29 20.68</td>
<td>10.65</td>
<td>41.26 28.05 19.19 14.90</td>
<td>5.32</td>
<td>19.31 0.68 0.50 0.45</td>
<td>0.59</td>
</tr>
<tr>
<td>PSMNet-vpp</td>
<td>✓ ✓</td>
<td>42.24 30.66 24.01 20.47</td>
<td>10.51</td>
<td>66.21 27.18 19.60 15.54</td>
<td>4.49</td>
<td>18.96 3.91 1.56 2.35</td>
<td>0.84</td>
</tr>
<tr>
<td>PSMNet-gd [51]</td>
<td>✓ ✓</td>
<td>26.30 16.08 13.06 11.60</td>
<td>6.11</td>
<td>22.26 10.52 6.96 5.36</td>
<td>1.63</td>
<td>10.83 2.01 1.46 1.30</td>
<td>0.53</td>
</tr>
<tr>
<td>PSMNet-pp</td>
<td>✓ ✓</td>
<td>23.52 13.61 10.74 9.73</td>
<td>4.28</td>
<td>17.76 12.37 7.73 5.22</td>
<td>1.79</td>
<td>12.28 2.47 1.86 1.60</td>
<td>0.52</td>
</tr>
<tr>
<td>PSMNet-gd-ft [51]</td>
<td>✓ ✓</td>
<td>25.87 14.92 11.87 10.43</td>
<td>6.19</td>
<td>21.21 10.17 6.78 5.26</td>
<td>1.66</td>
<td>2.86 1.40 1.21 1.11</td>
<td>0.38</td>
</tr>
<tr>
<td>PSMNet-pp-ft</td>
<td>✓ ✓</td>
<td>24.71 12.81 9.12 7.29</td>
<td>3.94</td>
<td>2.81 3.95 5.75 4.10</td>
<td>1.33</td>
<td>2.99 0.80 0.52 0.42</td>
<td>0.24</td>
</tr>
<tr>
<td>LiDARStereoNet [51]</td>
<td>✓ ✓</td>
<td>33.35 12.05 10.52 9.35</td>
<td>5.09</td>
<td>16.24 8.87 8.73 5.40</td>
<td>1.97</td>
<td>1.98 1.46 1.36 1.29</td>
<td>0.34</td>
</tr>
<tr>
<td>CCvNet [51]</td>
<td>✓ ✓</td>
<td>30.12 12.49 7.54 5.58</td>
<td>2.27</td>
<td>20.88 7.63 4.47 3.28</td>
<td>1.14</td>
<td>17.81 4.09 2.18 1.26</td>
<td>0.66</td>
</tr>
<tr>
<td>RAFT-Stereo-tp</td>
<td>✓ ✓</td>
<td>23.24 13.65 12.48 10.66</td>
<td>5.97</td>
<td>20.65 10.48 7.55 5.55</td>
<td>1.31</td>
<td>2.84 1.44 1.06 0.74</td>
<td>0.45</td>
</tr>
<tr>
<td>RAFT-Stereo-gd [51]</td>
<td>✓ ✓</td>
<td>8.04 5.30 4.35 3.81</td>
<td>2.01</td>
<td>6.94 4.26 3.33 2.79</td>
<td>0.72</td>
<td>1.30 0.83 0.65 0.54</td>
<td>0.12</td>
</tr>
<tr>
<td>RAFT-Stereo-pp</td>
<td>✓ ✓</td>
<td>15.22 8.02 5.97 5.05</td>
<td>3.97</td>
<td>12.71 6.36 4.63 3.82</td>
<td>1.21</td>
<td>3.32 1.26 1.00 0.77</td>
<td>0.20</td>
</tr>
<tr>
<td>RAFT-Stereo-gd-ft [51]</td>
<td>✓ ✓</td>
<td>7.12 4.73 3.95 3.51</td>
<td>1.95</td>
<td>6.40 4.00 3.20 2.76</td>
<td>0.83</td>
<td>1.01 0.74 0.64 0.58</td>
<td>0.14</td>
</tr>
<tr>
<td>RAFT-Stereo-pp-ft</td>
<td>✓ ✓</td>
<td>6.78 3.29 2.35 1.83</td>
<td>0.91</td>
<td>6.45 3.14 2.25 1.82</td>
<td>0.64</td>
<td>0.94 0.59 0.46 0.38</td>
<td>0.14</td>
</tr>
<tr>
<td>RAFT-Stereo-gd-tr [51]</td>
<td>✓ ✓</td>
<td>5.07 4.14 3.63 3.33</td>
<td>1.51</td>
<td>3.85 3.05 2.39 2.01</td>
<td>0.66</td>
<td>0.71 0.57 0.51 0.47</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Table 2: Comparison with existing methods. Results on Midd-14, Midd-21, ETH3D. Networks trained on synthetic data.

4.3. Ablation Study

We start by studying the different pattern variants and components under different settings. Tab. 1 summarizes this ablation study, conducted on Midd-A at full resolution with PSMNet [8], RAFT-Stereo [41] and rSGM. For each stereo pair, we randomly sample 5% sparse depth points from the dense ground-truth. Simply enabling virtual projection with pointwise virtual patterns (ii) and (iii), without explicit handling of occlusions, gives a massive boost in performance compared to the baseline, dropping the error rate to half for RAFT-Stereo (11 to 5%) and PSMNet (29 to 15%), while also reducing the error rate from 34 to 20% for rSGM. Adding alpha-blending with pointwise patterns yields similar, yet slightly worse results across all methods. Nonetheless, projecting patterns on 3 × 3 patches in conjunction with alpha-blending enables additional and consistent improvements to all methods with virtual pattern (iv) the most effective except for rSGM, which is not surprisingly more effective with pattern (vii) since it builds the cost volume based on a pointwise cost function. By handling occlusions thanks to our heuristic, we obtain some further improvements when neglecting projection (“NO” strategy). Finally, enabling occlusion handling through the “FGD” strategy yields the best results for all methods, with slight differences among them with the four patch-based virtual patterns. Considering the outcome of the ablation, despite the slightly worse accuracy, we conduct the remaining experiments with virtual pattern vi) on 3 × 3 patches, because of its negligible computation overhead with respect to histogram-based projection, while enabling at the same time alpha-blending and “FGD” occlusion handling – i.e., in yellow in the table.

4.4. Comparison with Existing Approaches

We now assess the performance of VPP and guided stereo. If not differently specified, we randomly sample 5% depth points from ground-truth, as in [51].

**Guided/VPP variants.** Give the flexibility of both VPP and guided stereo, we evaluate their application to stereo networks under three different settings, respectively:

- Without retraining the stereo network (-gd/-vpp)
- After a brief fine-tuning of the pre-trained model, enhanced by guided or VPP frameworks (-gd-ft/-vpp-ft)
- By training from scratch a new model, enhanced by guided or VPP (-gd-tr/-vpp-tr)

For -ft variants, one epoch of finetuning is carried out on FlyingThings, with learning rate 1e-4 and 1e-5 for PSMNet and RAFT-Stereo, respectively. For -tr variants, PSMNet and RAFT-Stereo are trained for 10 and 20 epochs, with learning rates 1e-3 and 1e-4, respectively. In any case, PSMNet and RAFT-Stereo process 384 × 512 and 360 × 720 crops, respectively, with batch size 2 on a single 3090 GPU.

The three variants allow for evaluating guided stereo and VPP when deployed with the least effort – i.e., by simply taking a pre-trained stereo network off the shelf – or with deeper intervention by the developer, either through a short
fine-tuning on the pre-existing model or, with major efforts, by retraining it from scratch.

Comparison on Middlebury/ETH3D. Tab. 2 reports the outcome on Midd-14, Midd-21 and ETH3D datasets using models trained on synthetic data only or the rSGM algorithm. We can immediately notice one of the most prominent figures of merit of our proposal, i.e., the remarkable ability to largely boost cross-domain generalization on all datasets without any retraining of the pre-existing model (-vpp). On the contrary, guided stereo improves the results only marginally under this setting (-gd), while CCVNNorm and LidarStereoNet need to be trained from scratch to process depth points (i.e., they are concatenated to RGB images). Accordingly, VPP is the undisputed winner in boosting the accuracy of stereo networks taken off the shelf.

By shortly fine-tuning the original networks to take advantage of sparse depth data is beneficial for both guided stereo (-gd-ft) and VPP (-vpp-ft) although the latter frequently, and always with RAFTStereo, outperforms the former even without any fine-tuning (-vpp), while training from scratch the networks to exploit the depth data with guided stereo (-gd-tr) results, sometimes, negligibly better than with VPP (-vpp-tr). This fact emphasizes further how VPP is much less training dependent, which is remarkable. Moreover, acting at the image level (VPP) is much less training dependent, which is remarkable.

Depth Sparsity vs Accuracy. Fig. 6 shows the error rate (> 2) on Midd-14 varying the density of sparse depth points from 0% to 5%. We can notice that VPP generally reaches almost optimal performance with a meagre 1% density and, except few cases in the -tr configurations with some higher density, achieves much lower error rates.

VPP + Existing Methods. Since our approach can be

Table 3: Combining VPP with existing methods. Results on Midd-14, networks trained on synthetic data.

<table>
<thead>
<tr>
<th>Model</th>
<th>Depth Points</th>
<th>Train</th>
<th>Test</th>
<th>&gt; 1</th>
<th>&gt; 2</th>
<th>&gt; 3</th>
<th>&gt; 4</th>
<th>avg. (px)</th>
</tr>
</thead>
<tbody>
<tr>
<td>rSGM [63]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.50</td>
</tr>
<tr>
<td>rSGM-gd</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
<tr>
<td>rSGM-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>9.94</td>
</tr>
<tr>
<td>rSGM-gd-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>9.65</td>
</tr>
<tr>
<td>PSMNet [8]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.50</td>
</tr>
<tr>
<td>PSMNet-gd</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
<tr>
<td>PSMNet-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>9.94</td>
</tr>
<tr>
<td>PSMNet-gd-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>9.65</td>
</tr>
<tr>
<td>LidarStereoNet-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
<tr>
<td>CCVNNorm [73]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.50</td>
</tr>
<tr>
<td>RAFT-Stereo [41]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
<tr>
<td>RAFT-Stereo-gd</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
<tr>
<td>RAFT-Stereo-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
<tr>
<td>RAFT-Stereo-gd-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
</tbody>
</table>

Table 4: Results with fine-tuned models. Results on Midd-21, after fine-tuning on Midd-14.

<table>
<thead>
<tr>
<th>Model</th>
<th>Depth Points</th>
<th>Train</th>
<th>Test</th>
<th>&gt; 1</th>
<th>&gt; 2</th>
<th>&gt; 3</th>
<th>&gt; 4</th>
<th>avg. (px)</th>
</tr>
</thead>
<tbody>
<tr>
<td>rSGM [63]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.50</td>
</tr>
<tr>
<td>rSGM-gd</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
<tr>
<td>rSGM-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>9.94</td>
</tr>
<tr>
<td>rSGM-gd-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>9.65</td>
</tr>
<tr>
<td>PSMNet [8]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.50</td>
</tr>
<tr>
<td>PSMNet-gd</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
<tr>
<td>PSMNet-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>9.94</td>
</tr>
<tr>
<td>PSMNet-gd-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>9.65</td>
</tr>
<tr>
<td>LidarStereoNet-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
<tr>
<td>CCVNNorm [73]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.50</td>
</tr>
<tr>
<td>RAFT-Stereo [41]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
<tr>
<td>RAFT-Stereo-gd</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
<tr>
<td>RAFT-Stereo-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
<tr>
<td>RAFT-Stereo-gd-vpp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.35</td>
</tr>
</tbody>
</table>

Table 5: Experiments on KITTI. Results on the 142 split [13] with raw LiDAR. Networks trained on synthetic data.
Table 6: VPP with off-the-shelf networks. Results on Midd-14, Midd-21, ETH3D and KITTI. * uses $\alpha = 0.2$ for blending.

![RAFTR-Stereo](image1.png)  ![RAFTR-Stereo-vpp](image2.png)  ![RAFTR-Stereo](image3.png)  ![RAFTR-Stereo-vpp](image4.png)

Figure 7: Performance by RAFT-Stereo-vpp on thin objects. Most fine details are preserved by VPP.
References


