Neural Disparity Refinement

Fabio Tosi, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, *Member, IEEE*, Samuele Salti, Stefano Mattoccia, *Senior Member, IEEE*, and Luigi Di Stefano, *Member, IEEE*

***Abstract*—We propose a framework that combines traditional, hand-crafted algorithms and recent advances in deep learning to obtain high-quality, high-resolution disparity maps from stereo images. By casting the refinement process as a continuous feature sampling strategy, our neural disparity refinement network can estimate an enhanced disparity map at any output resolution. Our solution can process any disparity map produced by classical stereo algorithms, as well as those predicted by modern stereo networks or even different depth-from-images approaches, such as the COLMAP structure-from-motion pipeline. Nonetheless, when deployed in the former configuration, our framework performs at its best in terms of zero-shot generalization from synthetic to real images. Moreover, its continuous formulation allows for easily handling the *unbalanced* stereo setup very diffused in mobile phones.**

***Index Terms*—Stereo matching, deep learning, domain generalization.**

I. Introduction

DEPTH estimation, from either images or custom sensors, represents one of the cornerstones upon which perception builds. To accomplish this task, two main imaging strategies have been developed through the years, consisting of *active* and *passive* technologies. Active technologies use specialized sensors to measure the distance between an emitter and an object, such as by measuring the time-of-flight of a signal or the deformation of a pattern. On the other hand, passive technologies use standard images and custom algorithms to compute depth cues. Passive technologies are generally cheaper and more flexible because they rely on inexpensive cameras, while active technologies are more expensive and may be limited by factors such as the environment and the minimum and maximum working range. Moreover, active technologies require additional energy to power their emitters, making them less efficient for edge platforms or mobile phones, which are increasingly equipped with multiple cameras and would benefit from an image-based solution coupled with efficient processing.

Among passive techniques, stereo [1] represents the cheapest and most widespread setup, reignited recently by deep learning [2]. Specifically, the stereo pipeline sketched by Scharstein and Szelinski [1] is nowadays implemented by means of *differentiable* components – i.e., 2D [3] or 3D [4] convolutional layers, correlation operators [3], [5] and, more recently, attention modules [6] – building end-to-end stereo architectures. However, an important factor in training these models to their best potential is the availability of large quantities of stereo pairs labeled with per-pixel disparities. Obtaining this type of data in practice can be costly, as it requires active sensors and manual cleaning of the collected data. As such, generating synthetic images using graphic engines such as Blender [3], [6] or Unreal [7] is a cheaper alternative, but it comes with its challenges. In particular, the *shift* that occurs when moving from synthetic to real images can diminish the overall accuracy of trained stereo models.

Another challenge that many stereo networks face is handling high-resolution images. Processing images with several megapixels can be difficult for several reasons. First, the memory requirements for 3D networks (height, width, and disparity search range) become prohibitively high. Second, most architectures have a relatively small receptive field, which can make the features extracted from high-resolution images insufficiently specific for matching. As a result, most stereo networks process downsampled pairs of high-resolution images, and then upsample the output disparity map, which leads to a loss of accuracy.

In this paper, we present a simple solution for estimating sharp and accurate depth maps from images. Acknowledging that traditional stereo algorithms already produce disparity maps of medium accuracy, we focus on designing a *Neural Disparity Refinement* network capable of enhancing the quality of these initial estimates. Specifically, our model leverages the reference RGB image to guide the refinement of the processed disparity map's structure, as well as to super-resolve it thanks to its continuous feature sampling strategy. This strategy allows us to produce high-quality, high-resolution disparities even when processing downsampled stereo images benefiting from the inherent robustness to domain shifts of traditional algorithms. As a result, our model, which is trained on synthetic datasets only, outperforms almost all end-to-end stereo networks in terms of zero-shot generalization accuracy. Nonetheless, our approach is a general and operates on heterogeneous stereo algorithms, both traditional and deep learning-based, and allows us to achieve more accurate results with sharp depth discontinuities, as shown in Fig. 1.

This work builds on our previous research on neural disparity refinement [8], with improvements to the overall framework and additional testing on several use cases. Specifically, we have updated our model in the following ways:

- We exhaustively study different configurations of our architecture. Specifically, we run experiments with various popular backbones to find the one that best fits our purpose.

Manuscript received 31 July 2023; revised 18 March 2024; accepted 5 June 2024. Date of publication 7 June 2024; date of current version 5 November 2024. This work was supported by the Huawei Technologies Oy (Finland). Recommended for acceptance by P. Tan. *(Corresponding author: Fabio Tosi.)*

The authors are with the Department of Computer Science and Engineering, University of Bologna, 40136 Bologna, Italy (e-mail: fabio.tosi5@unibo.it; filippo.aleotti2@unibo.it; pierluigi.zama@unibo.it; m.poggi@unibo.it; samuele.salti@unibo.it; stefano.mattoccia@unibo.it; luigi.distefano@unibo.it).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3411292, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3411292

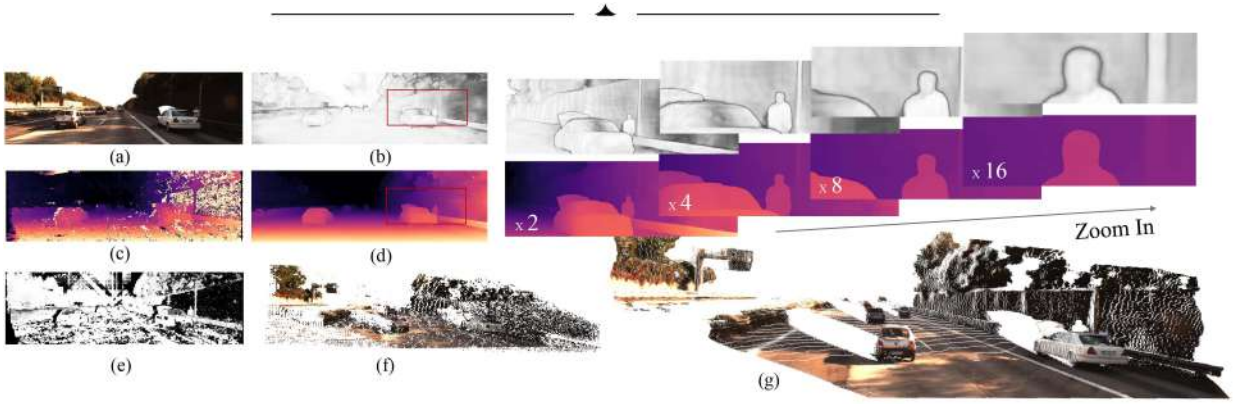

Fig. 1. *Outcome of our Neural Disparity Refinement framework.* Given an RGB image from the KITTI dataset (a) and a corresponding noisy depth map (c) computed using either stereo or multi-view images, our architecture estimates a refined depth map (d) with accurate and sharp edges at arbitrary, continuous output resolution based on a single training on synthetic data only. Optionally, our network predicts a confidence score for each point of the noisy input depth map (e) as well as the underlying aleatoric uncertainty (b). This allows us to produce a clean and consistent point cloud (g) compared to the initial 3D reconstruction obtained using traditional or deep learning-based methods (f).

- We extend our architecture to explicitly model, in addition, the confidence [9] of the noisy input disparity map.

Our contributions have allowed us to increase the accuracy originally achieved by our framework [8] as well as to tackle the crucial confidence estimation task. In summary, our neural disparity refinement network, trained on synthetic datasets only, enables us to:

- Refine raw disparity maps that have been generated by off-the-shelf traditional stereo algorithms or predicted by modern, end-to-end stereo networks, outperforming existing approaches [10], [11], [12].
- Achieve state-of-the-art accuracy in zero-shot generalization when processing disparity maps produced by hand-crafted stereo algorithms.
- Modelling the confidence of the input disparity on par – and often better – than state-of-the-art solutions specifically designed for this purpose [9].
- Super-solving the input disparity at any arbitrary resolution, enabling us to handle high-resolution stereo images (typically downsampled to reduce memory requirements) and deal with unbalanced stereo setups (commonly found in mobile phones).
- Improve the quality of other depth-from-images approaches beyond stereo algorithms, such as the popular structure-from-motion pipeline COLMAP [13].

Our source code is available at https://github.com/CVLAB-Unibo/neural-disparity-refinement.

II. Related Work

In this section, we review the literature about stereo matching from traditional approaches to deep models.

Traditional Stereo Matching. Traditional stereo algorithms, typically categorized into *local* and *global*, generally implement a subset of the four steps pipeline [1] consisting of: 1) matching cost computation, 2) cost aggregation, 3) disparity computation/optimization and, optionally, 4) disparity refinement. While *local* methods, also known as area-based algorithms, are mainly focused on the first two steps, *global* strategies mostly rely on step 3 by minimizing a global cost function that combines data and smoothness terms. While local algorithms are fast, global algorithms are generally more accurate but more computationally expensive. An intermediate point between accuracy and runtime is represented by the Semi-Global Matching (SGM) algorithm [14] that approximates the global solution by solving a 1D minimization problem along several paths (typically 8 or 16) across the image, reducing the computational costs compared to global matching techniques.

Learning for Stereo Pipeline. Recently, CNNs have been introduced to deal with stereo matching by Zbontar and LeCun [15], initially to replace the computation of hand-crafted matching cost functions with learning-based functions. While these approaches have achieved higher accuracy on main benchmarks, disparity maps computed using learned matching costs are still prone to errors in occluded regions, thin structures, and reflective surfaces. Therefore, they require post-processing operations to refine the resulting disparity maps. Among them, näive refinement practices typically involve left-right consistency check [16], hole filling [17], median filtering [18], [19], [20], [21], guided [22] or bilateral filtering [23] procedures. Others, instead, adopt non local-means filtering [24], dictionary-based strategies [25], filter forests [26] or leverage confidence estimation [9], [27] to determine the reliability of computed disparities and, possibly, refine them [28], [29]. With the advent of deep learning, deep neural networks have been used to detect, replace, and refine noisy disparities [10] or the refinement process has been formulated as a recurrent process [11].

Deep Stereo. Nowadays, end-to-end deep stereo matching networks have become the most popular and effective solution for the disparity estimation task. They can be mainly categorized into 2D architectures and 3D architectures. DispNet [3] represents the promotive network for more advanced deep architectures belonging to the first category [30], [31], [32], [33], [34], [35], [36], [37], while GC-Net [4] was the first model to propose

the use of an explicit 3D feature cost volume built using feature concatenation (or difference), on which more recent networks have been developed [38], [39], [40], [41], [42], [43], [44], [45], [46], [47]. Very recently, novel deep stereo networks exploit the iterative refinement paradigm proposed in the state-of-the-art optical flow network RAFT [48] to design architectures adapted for the stereo matching task [5], [6] or rely on Transformers [49], [50] to capture long-range contextual information aimed at improving disparity predictions on challenging regions. However, deep learning-based stereo methods rely heavily on expensive and hard-to-source ground-truth depth labels for training. As a result, they typically outperform traditional hand-crafted optimization schemes when a large amount of annotated data is available. In contrast, deep models that can learn directly from input stereo pairs alleviate the need for ground-truth depth annotations. Godard et al. [51] first proposed a self-supervised image-based loss function to train deep monocular and stereo models without ground-truth annotation. Since then, several approaches have been proposed to address this problem. Some of them exploited stereo video sequences and LSTM [52], others traditional stereo algorithms and confidence measures to detect highly reliable disparities from noisy depth maps [53], [54] or pre-trained stereo models [55]. Furthermore, other works take advantage relationship between optical flow and stereo [56] for the same purpose. More recent approaches, instead, propose to continuously adapt in a self-supervised manner a stereo network based on images captured by the camera during its deployment [32], [57], [58], [59].

Zero-Shot Generalization. A parallel line of research focuses on preserving depth accuracy across heterogeneous environments, assuming that information of the target domain is completely unavailable during training. The classical approach in the literature is to train deep stereo networks on synthetic datasets with available ground-truth disparity, such as the SceneFlow dataset [3], and test them on real-world scenes. However, deep stereo architectures are severely affected by the *domain-shift* issue if näively trained on images that are widely different from those at testing time. To cope with this issue, earlier approaches focused on learning domain-invariant features from hand-crafted matching functions on the input stereo pairs [60], on domain normalization operations [61] or sparse depth points from sensors [62], [63]. More recent works, instead, use contrastive features and stereo whitening losses [64] to better preserve stereo feature consistency across domains, leverage features learned on large-scale datasets (e.g. ImageNet) to graft them into the 3D cost volume [65], or exploit the Fisher information to restrict the shortcut-related information from being encoded from the input into the feature representations [66]. Other works, such as *Mono-for-Stereo* (MfS), tackle the domain shift problem by training deep stereo networks on stereo data synthesized from a large collection of monocular real-world RGB images. This goal can be achieved by employing depth maps predicted using a pre-trained monocular depth network [67], or by using neural radiance fields [68].

Confidence Estimation. Confidence measures can provide valuable information for improving the accuracy of 3D reconstruction results. Specifically, they can be used to identify areas of the scene that may be less reliable and need further refinement, as well as areas that are more likely to be accurate and can be used with greater confidence in downstream tasks. Recent trends in confidence measures for stereo matching are relevant to our work. For decades, many techniques have been proposed for estimating the reliability of disparity maps. Hu and Mordohai conducted an exhaustive survey on hand-crafted methods [69]. More recently, advances in deep neural networks have allowed for the development of machine and deep learning-based methods that can more accurately estimate confidence measures, as thoroughly studied in [9], [27].

Continuous Output Representation. Recently, there has been growing interest in utilizing continuous function representations across various domains, such as 3D reconstruction [7], [70], [71], [72], [73], [74], [75], [76], [77], texture estimation [78], image synthesis [79], [80], semantic segmentation [81], image super-resolution [82] and correspondence matching [83]. Likewise, our proposed work also adopts a continuous representation of the output, highlighting the advantages and effectiveness of continuous function representations in the disparity refinement task.

III. Proposed Framework

Our Neural Disparity Refinement framework, illustrated in Fig. 2, processes a raw, noisy disparity map and its corresponding RGB image to yield an enhanced prediction with sharper depth discontinuities and (optionally) higher resolution. It consists of a feature extraction backbone, implemented as a standard convolutional neural network, which processes the RGB image and noisy depth map separately, and Multi-Layer Perceptron (MLP) heads that predict the final refined output. Specifically, the designed MLPs query interpolated feature vectors extracted by the backbone at continuous pixel locations, allowing for estimating outputs at any arbitrary spatial resolution. The whole framework is trained in an end-to-end manner, optimizing for the parameters of any components at once. In the following subsections, we will explain in detail the design and role of each module.

A. Neural Disparity Refinement Network

During the forward pass, our model performs three main steps: 1) extracting features from the disparity map and RGB image, 2) sampling and interpolating features, and 3) inferring disparity/confidence and enhancing subpixel accuracy. The basic blocks of our model are shown in Fig. 2.

Features Extractor. At the very beginning of our framework, an input RGB image $\mathcal{I}$ and its corresponding disparity map $\mathcal{D}$, are forwarded to our model. $\mathcal{D}$ is the raw disparity map we aim at refining by exploiting the guidance from $\mathcal{I}$, and may come from any multi-view stereo or binocular (balanced/unbalanced) stereo algorithm, either hand-crafted or learned. Two separate encoders $\phi_{\mathcal{I}}$ and $\phi_{\mathcal{D}}$ – implemented by using a standard backbone from the literature – extract multi-scale features denoted here as $\boldsymbol{\mathcal{F}}_{\mathcal{I}}$ and $\boldsymbol{\mathcal{F}}_{\mathcal{D}}$, from the RGB image and the noisy disparity map respectively. Then, a late-fusion decoder, i.e., Δ in Fig. 2, blends the features from the two encoders and gradually upsample

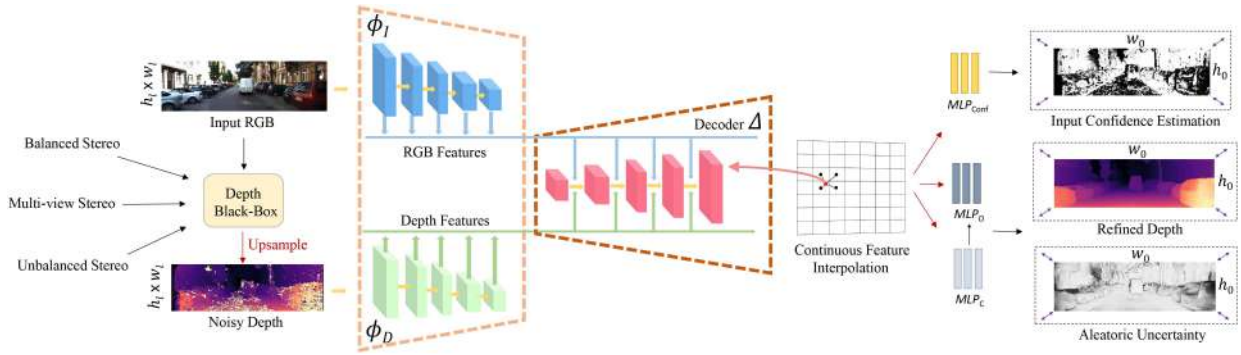

Fig. 2. *Architecture Overview.* Given noisy depths $\mathcal{D}$ pre-computed by any existing depth blackbox, specifically tailored for either the multi-view stereo or the standard (balanced/unbalanced) stereo matching task, our network aims to estimate a refined depth map $\widetilde{\mathcal{D}}$ at any arbitrary spatial resolution. First, we extract deep features from both $\mathcal{I}_l$ and $\mathcal{D}$ using two separate convolutional branches, $\phi_{\mathcal{I}}$ and $\phi_{\mathcal{D}}$, which are combined by a decoder Δ. Then, at each continuous 2D location in the $\mathcal{I}_l$ image domain, we interpolate features across the levels of Δ and feed them into a disparity estimation module, consisting of two MLPs, namely MLP_C and MLP_O, which predict an integer disparity value and a sub-pixel offset, respectively. Furthermore, our network can be optionally extended to predict a confidence score for each point of the noisy input depth map using an additional MLP, namely MLP_{conf}.

them back to the original input resolution. At each single level l in Δ, the features from the very same level in the encoder, namely $\boldsymbol{\mathcal{F}}^l_{\mathcal{I}}$ and $\boldsymbol{\mathcal{F}}^l_{\mathcal{D}}$, are aggregated by mean of a channel-wise sum operation and combined with the upsampled features from the previous decoder level, namely $\boldsymbol{\mathcal{F}}^{l-1}_{\Delta}$, by means of skip-connections.

Features Sampling and Interpolation. The extracted features from the decoder are then used to infer the refined disparity map. This is achieved through the use of multiple MLPs that are queried with features sampled at continuous 2D pixel coordinates. As previously mentioned, we use bilinear interpolation to obtain the corresponding features for a given 2D continuous coordinate (x, y), by sampling them at the four nearest discrete locations, similar to the method described in [7]. This process is performed at any level l from the decoder. Sampling and interpolating features from the decoder, along with predicting the refined disparity using MLPs allows our model to refine the input disparity map at any desired resolution. By continuously sampling and interpolating features from the decoder at different resolutions, we can predict the refined disparity at any continuous 2D location rather than being limited to predicting it only at discrete positions. This flexibility in predicting the refined disparity at any desired resolution is achieved by using bilinear interpolation to obtain the features at continuous 2D coordinates, and then concatenating and forwarding them to the MLPs to make the final prediction. This approach enables our model to produce high-resolution, finely-detailed disparity maps, as demonstrated in our experimental results.

Disparity Prediction and Subpixel Enhancement. To prevent disparity over-smoothing, which can be a problem for deep stereo networks that infer disparities through regression [84], we split the final prediction into two separate components: a discrete component and a subpixel component. Indeed, over-smoothing can have severe consequences on downstream applications processing 3D data due to the bleeding artifacts [7], [84] that can be generated at depth discontinuities. This issue is mainly caused by the multi-modal distribution that is learned for pixels located near depth discontinuities. This distribution is encoded either within the cost volume in 3D networks, which causes the soft-argmax operator to infer a disparity between the foreground and background modes, or within the final regression layer in 2D networks. To deal with it, we cast disparity estimation into two distinct sub-processes deploying two MLPs. Given features sampled at any 2D location, the former MLP, namely MLP_C predicts a probability distribution P over a discrete 1D disparity interval $Z = [z_0, z_1, .., z_{d_{\max}-1}]$, thus tackling disparity estimation as a classification problem, while the latter, namely MLP_O, is in charge of regressing a sub-pixel displacement to be added to the discrete disparity predicted by MLP_C. In particular, both MLPs process features $\boldsymbol{\mathcal{F}}_{\Delta}$ – defined as the concatenation of features $\boldsymbol{\mathcal{F}}^l_{\Delta}$ sampled from any level l – with MLP_O being also fed with the discrete disparity being predicted by MLP_C. Finally, we define the final, refined disparity value $\widetilde{\mathcal{D}}$ as the sum between the two predictions by the MLPs:

$$\widetilde{\mathcal{D}} = \text{argmax}(\text{MLP}_C(\boldsymbol{\mathcal{F}}_{\Delta})) + \text{MLP}_O(\boldsymbol{\mathcal{F}}_{\Delta}) \tag{1}$$

with MLP_C trained to predict a discrete disparity value by the minimization of a cross-entropy loss, while MLP_O being optimized to infer an offset in the range $[-1, 1]$, by adopting an $\mathcal{L}_1$ loss. Accordingly, the activations placed after the last layer in MLP_C and MLP_O are the Softmax and Tanh operators, respectively. Given $\mathcal{D}^*$ as the ground-truth disparity, the final loss function computed during training, $\mathcal{L}$, is obtained as the sum of two terms:

$$\begin{aligned}\mathcal{L} = & -\mathcal{N}(\mathcal{D}^*, \sigma) * \log\left(\text{MLP}_C(\boldsymbol{\mathcal{F}}_{\Delta})\right) \\ & + \left|\text{MLP}_O(\boldsymbol{\mathcal{F}}_{\Delta}) - \mathcal{D}^*_s\right|\end{aligned} \tag{2}$$

with $\mathcal{N}(\mathcal{D}^*, \sigma)$ being a Gaussian distribution centered at $\mathcal{D}^*$, with σ being set to $\sqrt{2}$, and $\mathcal{D}^*_s$ being the difference between $\mathcal{D}^*$ and the integer disparity predicted by MLP_C, respectively:

$$\mathcal{D}^*_s = \mathcal{D}^* - \text{argmax}\left(\text{MLP}_C(\boldsymbol{\mathcal{F}}_{\Delta})\right) \tag{3}$$

i.e., the offset added to bring the discrete disparity predicted by MLP_C closer to $\mathcal{D}^*$. As such, the latter term in (2), in charge of subpixel enhancement of the final disparity value, is minimized only if $\mathcal{D}_s^*$ results in $[-1, 1]$. Notice that our model also allows for capturing the underlying aleatoric uncertainty of the predicted refined depth map by evaluating the entropy of the discrete disparity distribution predicted by MLP_C. More specifically, we compute the output uncertainty as:

$$H = \sum_{d=0}^{d_{\max}-1} P(z_d)\log(P(z_d)) \tag{4}$$

Input Confidence Estimation. An additional multi-layer perceptron (MLP), named MLP_{conf}, can be used to infer the confidence of each point in the input disparity map from the same features used for disparity estimation. This choice enables us to maintain our continuous formulation, allowing confidence estimation at arbitrary spatial resolutions. Specifically, this module outputs a single value normalized to the range [0,1] using a sigmoid activation function. During training, an additional loss term $\mathcal{L}_{\text{conf}}$ is calculated as follows:

$$\mathcal{L}_{\text{conf}} = -\mathcal{C}^* * \log\left(\text{MLP}_{\text{conf}}(\boldsymbol{\mathcal{F}}_{\Delta})\right) \tag{5}$$

with $\mathcal{C}^*$ being the binary ground-truth confidence map, obtained by setting a threshold γ on the absolute difference between the input and ground-truth disparities $\mathcal{D}$ and $\mathcal{D}^*$, respectively. A value of 1 (indicating high confidence) is assigned to pixels with a difference lower than γ, 0 to the remaining ones.

IV. Experimental Results

In this section, we present the results of our extensive experimental evaluation. First, we describe the implementation details of our framework. Next, we introduce the datasets used in our experiments. Finally, we report our findings.

A. Implementation Details

We begin by outlining the implementation details of our framework, including the architecture, training schedule, and evaluation metrics used in our experiments.

Architecture. Our refinement architecture is compatible with any feature extractor from the literature. In our experiments, we assess the impact of various backbones, including standard convolutional networks (such as VGG [85] and ResNet [86]) and more advanced structures based on Transformers. We use the official code provided by the authors for all the adopted feature extractors. For the decoder, we use the same structure described in [8], based on multiple upsampling layers. Any of MLP_C, MLP_O and MLP_{conf} is implemented as a Multi-Layer Perceptron (MLP) following [7]. More specifically, the number of neurons is $(N, 1024, 512, 256, 128, d\max)$ for MLP_C, $(N+1, 128, 64, 1)$ for MLP_O and $(N+1, 128, 128, 128, 1)$ for MLP_{conf}, where N represents the total number of features extracted from the final 4 upsampling modules within the decoder. These modules are comprised of convolutional layers followed by nearest-neighbor upsampling. Unless otherwise stated, we set the maximum disparity value $d\max$ to 256. Layers of each MLP are followed by the Sine operator, except the last one in which we use Softmax, Tanh and Sigmoid functions for MLP_C, MLP_O and MLP_{conf}, respectively.

Training Protocol. Our refinement framework is implemented in PyTorch and all models are trained from scratch using a single NVIDIA 3090 GPU, unless otherwise stated. We use Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for optimization. During training, we use batch size B and randomly crop images of size 384×384 from the input RGB and noisy disparity images (both normalized in the range [0,1]). We also sample K random training points from each crop. For ablation studies using the VGG-13 backbone, similar to our previous work, we set $B = 8$ and $N = 30,000$. For the experiments with other backbones, we optimize N and B to optimize GPU utilization and achieve the best training performance for each, given the same GPU memory amount (24 GB in our default configuration, with a single 3090 GPU). Specifically, we fix $N = 5,000$ and set the batch size B to be as large as possible to fit the available memory budget.

During training, we dynamically generate noisy disparity maps on-the-fly by randomly alternating between two traditional yet popular stereo matching algorithms: SGM [14] and AD-Census [87]. This approach enables us to leverage the complementary strengths of both algorithms and create a more diverse and robust training dataset. Moreover, the seamless integration of these algorithms within the OpenCV framework enables us to compute disparity maps in real-time and on-the-fly, contributing significantly to the effectiveness of our training phase. For both of them, we randomly select different parameter configurations during training and use 256 as the maximum disparity hypothesis $d\max$. For SGM, we adopt [3,5,7] as block size b, P1 as $2 \times b \times b$, P2 as $K \times b \times b$ where $K \in [32, 64, 96]$ and, if enabled, we set the left-right consistency check threshold as $[1, 2, 5]$. For AD-Census, we randomly select the block size in the range [7,21] and a uniqueness ratio value in [0,15]. Furthermore, we randomly feed as input of our network an augmented version of the ground-truth disparity by sequentially downsampling and upsampling it using a random scale factor belonging to the range $[1, 2, 4, 8]$ and different interpolation methods, such as nearest-neighbor, bilinear, bicubic and resampling using pixel area relation. This allows to improve the network's capability to handle unbalanced stereo input. We feed our network with disparity maps computed by SGM with probability 0.45, AD-Census with probability 0.45 while the augmented ground-truth with probability 0.1. In any case, the input disparities - and, accordingly, the ground-truth - are further augmented, i.e., scaled to have zero translation and unit scale, normalized with respect to the minimum and maximum of the considered ground-truth and, finally, rescaled again by selecting randomly and uniformly a scale factor in $[d\min, d\max]$, where $d\min = 20$. This procedure allows us to cover the entire disparity distribution regardless of the dataset adopted at training time. We further augment the training procedure by flipping the input disparities and the corresponding RGB/ground-truth images horizontally and vertically. We also augment the input RGB image using several augmentation functions such as RGB shift, channel dropout/shuffle, histogram equalization, random

brightness, contrast and gamma, motion and median blur, Gaussian noise, image compression and conversion to grayscale. For the input confidence estimation task, we set $\gamma = 2$.

Evaluation Metrics. To validate our models and compare them to other existing frameworks on popular stereo benchmarks, we use the standard end-point-error (EPE) metric computed by averaging the absolute difference between the predicted disparity maps and the ground-truth. Furthermore, we compute the percentage of pixels having EPE higher than τ pixels - referred to in the literature as $bad\tau$ - as well as the Soft-Edge-Error (SEE) metric [84], aimed at assessing the sharpness of the predictions at depth discontinuities. Unless specified otherwise, we set $\tau = 3$ for KITTI, $\tau = 2$ for Middlebury and $\tau = 1$ for ETH3D and evaluate the performance of our method on both occluded and non-occluded areas. For an exhaustive comparison to other frameworks, on the KITTI dataset, we also report the $D1$ error defined as the percentage of pixels where the 3-px error is greater than 5% of the ground-truth. As in our previous work [8], we fix the size of ground-truth patches to 5×5 for the SEE evaluation.

B. Datasets

Now, we introduce the datasets used in our experiments. It is worth stressing that, in our experiments, we adopt synthetic datasets only for the training phase.

1) Training: UnrealStereo4K. The UnrealStereo4K dataset [7] is a synthetic binocular stereo dataset depicting photo-realistic environments with available ground-truth disparities. Created using the popular game engine Unreal Engine, the entire dataset consists of 8,000 stereo pairs at 3840×2160 resolution. In our experiments, we also consider a low-resolution variant, dubbed UnrealStereo4K-Q, obtained by downsampling both images and disparities by a factor of 4.

SceneFlow. The SceneFlow dataset [3] is a collection of images containing more than 39,000 stereo frames at 960×540 resolution with corresponding dense ground-truth disparities. Similarly to our work [8], we use 22,340 stereo pairs for training, 50 for validation, and 387 for testing.

Virtual KITTI 2. The Virtual KITTI 2 dataset [89] consists of 5 sequence clones from the KITTI tracking benchmark, providing different variants of these sequences, such as modified weather conditions or camera configurations.

CREStereo Dataset. The CREStereo dataset [6] is a collection of stereo images rendered at 1920×1080 using Blender, with ground-truth disparity. It features scenes similar to those in the Sceneflow dataset, but with higher object complexity and resolution. Like UnrealStereo4K, we conduct experiments using a lower resolution variant, dubbed CREStereo-H, obtained using a downsampling factor of 2.

2) Testing: KITTI. The KITTI dataset is a real-world stereo dataset depicting autonomous driving scenarios captured at 1280×384 resolution, with sparse ground-truth depth maps collected using a LiDAR sensor. For evaluation purposes, we consider both the KITTI 2015 training dataset [90], which features 200 stereo pairs, and the KITTI 2012 training dataset [91], which contains 194 stereo images.

TABLE I
ABLATION STUDY – OUTPUT REPRESENTATION

Input	Network	Output Repr.	EPE	SEE	bad2	FPS	Size (Mb)
AD-Census [87]	-	Initial Depth	24.68	21.78	46.23	-	-
	FC	L_1	2.25	4.00	16.25	53.8	22.53
	CNN+MLP	L_1	1.86	3.14	13.94	17.70	22.80
	CNN+MLP	Bimodal [7]	1.66	**1.47**	9.37	17.63	22.80
	CNN+MLP	Ours	**1.53**	1.48	**8.49**	13.70	22.88
C-CNN [88]	-	Initial Depth	13.46	10.84	27.09	-	-
	FC	L_1	2.33	4.65	13.85	53.8	22.53
	CNN+MLP	L_1	1.68	2.86	10.54	17.70	22.80
	CNN+MLP	Bimodal [7]	1.64	1.38	7.93	17.63	22.80
	CNN+MLP	Ours	**1.42**	**1.36**	**7.11**	13.70	22.88
SGM [14]	-	Initial Depth	9.51	8.51	23.34	-	-
	FC	L_1	1.54	2.77	10.22	53.8	22.53
	CNN+MLP	L_1	1.37	2.63	8.86	17.70	22.80
	CNN+MLP	Bimodal [7]	1.25	**1.17**	6.08	17.63	22.80
	CNN+MLP	Ours	**1.19**	1.26	**5.80**	13.70	22.88

We report the results obtained by our network when trained using a standard L_1, a bimodal mixture representation [7] or our proposed loss. FC indicates Fully-Convolutional, while CNN+MLP refers to an encoder-decoder CNN architecture combined with an MLP designed to estimate the disparity.
The bold values indicate the best results.

Middlebury v3. The Middlebury v3 dataset [92] is a high-resolution ($\sim$ 5Mpx) stereo dataset depicting indoor scenes under controlled lighting conditions, containing 15 stereo pairs at Full (F), Half (H), and Quarter (Q) resolution, annotated with semi-dense ground-truth disparities.

ETH3D. The ETH3D stereo dataset [93] includes a mix of indoor and outdoor scenes, with a total of 27 grayscale low-resolution ($\sim$ 0.4 Mpx) stereo pairs and corresponding ground-truth disparity maps.

C. Ablation Study

We begin our experiments by conducting ablation studies to measure the impact of design choices, augmentation strategies, and training datasets on the performance of our final model. Experiments for the ablation study utilize the SceneFlow dataset, employing the VGG-13 backbone as the default feature extractor.

Output Representation. Table I presents the outcomes derived from raw disparity maps estimated by AD-Census [87], C-CNN [88], or SGM [14] obtained by training our framework with different losses, leading to different representations used to produce refined disparity maps: a standard $\mathcal{L}_1$ loss, the bimodal loss proposed by SMDNets [7], and our strategy. Furthermore, while evaluating various loss functions, we investigate the impact of employing an MLP for disparity estimation, as opposed to a standard Fully Convolutional (FC) network configuration. In the FC network architecture, a final convolutional layer on top of the decoder is dedicated to estimating the refined disparity.

Upon thorough analysis of the table, it is evident that employing an MLP for disparity estimation consistently outperforms a fully convolutional network across all assessed algorithms when adopting the naïve $\mathcal{L}_1$ loss function. However, it is important to note that this enhanced accuracy accompanies a notable trade-off, resulting in a substantial decrease in runtime performance. Specifically, the Frames Per Second (FPS) rate decreases from 52.6 with the FC configuration to 17.54 using the CNN+MLP setup. Despite this compromise between accuracy and speed, we stress that our primary focus remains on achieving optimal disparity accuracy. This observation, coupled with the MLP's capability to estimate depth at any arbitrary resolution, directs

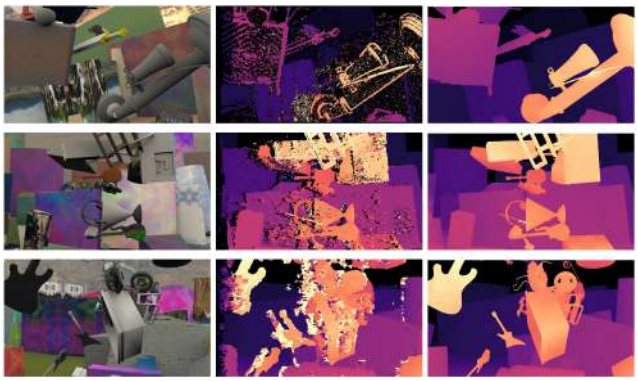

Fig. 3. *Qualitative Results – Stereo Refinement.* On each row, we show the RGB reference image and disparity maps estimated by traditional stereo algorithms and refined by our method. From top to bottom, examples using AD-Census [87], SGM [14], and C-CNN [88].

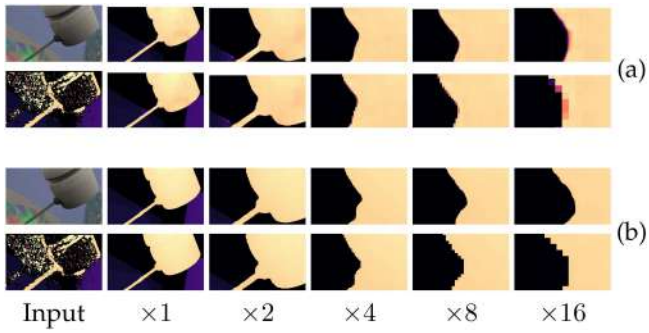

Fig. 4. *Upsampling Comparison*. We show disparity maps, upsampled by means of the continuous formulation (top) in our framework or nearest-neighbor interpolation (bottom), according to different scale factors (up to $\times 16$). (a) training with $\mathcal{L}_1$ loss, (b) training with our proposed loss.

TABLE II
ABLATION STUDY - EARLY VERSUS LATE FUSION

Input	Fusion	EPE	SEE	bad2
AD-Census [87]	Initial Depth	24.68	21.78	46.23
	Early	1.74	1.69	9.27
	Late	**1.53**	**1.48**	**8.49**
C-CNN [88]	Initial Depth	13.46	10.84	27.09
	Early	1.76	1.64	8.00
	Late	**1.42**	**1.36**	**7.11**
SGM [14]	Initial Depth	9.51	8.51	23.34
	Early	1.39	1.38	6.26
	Late	**1.19**	**1.26**	**5.80**

We report the results obtained by our network deploying a single encoder to process RGB and disparity inputs or two distinct encoders, respectively.
The bold values indicate the best results.

our subsequent analysis exclusively towards the CNN+MLP configuration. Considering this latter case from now on, it is worth noting that, for both the standard $\mathcal{L}_1$ and bimodal mixture output representations, a single instance of an MLP is responsible for regressing either the final refined disparity (for the former) or the five parameters of a univariate bimodal mixture distribution (for the latter), by modifying the last layer of our MLP_C and dismissing MLP_O. Notably, we can notice that all the trained models, using different output representations, significantly improve the accuracy of the considered noisy disparity maps computed using heterogeneous algorithms, as clearly shown in Fig. 3. This outcome indicates the effectiveness of our framework regardless of the stereo algorithm used to compute the initial disparity estimates, including stereo black-boxes never used at training time, such as the learning-based method C-CNN [88]. The standard disparity regression based on $\mathcal{L}_1$ produces the worst results among the three considered output representations. We can notice, instead, how our solution consistently achieves the lowest error rate with any threshold, as well as the lowest EPE, while slightly outperforming the bimodal output representation in terms of SEE when refining AD-Census and SGM disparity maps. Moreover, the table illustrates that the bimodal and $\mathcal{L}_1$ approaches, in the CNN+MLP configuration, operate at approximately 17 FPS, while our methodology processes at around 12 FPS. Despite our approach having a slightly lower FPS (attributed to using two MLPs), it maintains competitive processing speed while significantly improving the accuracy of disparity map refinement, especially in EPE and bad metrics. This evidence indicates our method's effectiveness in enhancing accuracy despite the slightly lower processing speed. Concerning the Size (Mb) metric, all methodologies present similar memory footprints, suggesting comparable sizes and similar memory storage requirements for deployment.

Furthermore, Fig. 4 illustrates the improvements in spatial resolution yielded by our continuous formulation compared to traditional nearest-neighbor interpolation, as well as the advantages of our proposed loss based on classification plus subpixel estimation over the standard $\mathcal{L}_1$ loss. Specifically, using a low-resolution ($0.5Mpx$) noisy disparity map as input, we refine and upsample the map using different upsampling factors, up to ($128Mpx$). It's crucial to highlight that both the nearest-neighbor interpolation and the standard $\mathcal{L}_1$ loss exhibit limitations. Nearest-neighbor interpolation introduces quantization artifacts, resulting in less precise and misaligned depth edges compared to the true edges in the scene. On the other hand, the conventional $\mathcal{L}_1$ loss introduces the issue of oversmoothing, blurring depth discontinuities and affecting the overall quality of the refined disparity maps. In contrast, our continuous formulation combined with the proposed loss consistently estimates depth maps with edges that are significantly sharper and far more precise. This improvement becomes even more pronounced at higher resolutions.

Early versus Late Fusion. Table II investigates the impact of two different configurations of our model. In particular, we compare 1) an early fusion method that directly concatenates the RGB image and the corresponding noisy disparity map before passing them through a standard encoder-decoder network, and 2) a late fusion model that employs two separate sub-networks to extract features from the RGB image using an RGB encoder network $\phi_{\mathcal{I}}$ and from the noisy disparity map using a disparity encoder network $\phi_{\mathcal{D}}$. The two sub-networks are then fused at intermediate layers through the sum operation. For both fusion strategies, we consider VGG-13 as the feature backbone [85]. The table shows that the use of specialized feature extractors leads to improved performance for all three considered stereo algorithm variants. This fact suggests that the late fusion strategy is more effective than the early fusion strategy for our disparity refinement framework. This finding can be attributed to the

TABLE III
ABLATION STUDY - NUMBER OF NEURONS

Input	MLP Size	EPE	SEE	bad2	FPS	Size (Mb)
AD-Census [87]	Initial Depth	24.68	21.78	46.23	-	-
	Tiny	1.75	1.66	10.01	22.41	22.62
	Small	1.68	1.59	9.86	19.24	22.69
	Normal [8]	**1.53**	**1.48**	8.49	13.70	22.88
	Large	1.58	1.51	**8.38**	9.30	23.54
C-CNN [88]	Initial Depth	13.46	10.84	27.09	-	-
	Tiny	1.73	1.61	8.23	22.41	22.69
	Small	1.49	1.43	7.96	19.24	22.69
	Normal [8]	1.42	1.36	7.11	13.70	22.88
	Large	**1.39**	**1.35**	**7.01**	9.30	23.54
SGM [14]	Initial Depth	9.51	8.51	23.34	-	-
	Tiny	1.34	1.41	7.05	22.41	22.69
	Small	1.25	1.43	6.97	19.24	22.69
	Normal [8]	**1.19**	**1.26**	**5.80**	13.70	22.88
	Large	**1.19**	1.29	5.91	9.30	23.54

Here we present the numbers obtained by varying the number of neurons in *MLPc* and *MLPo*.
The bold values indicate the best results.

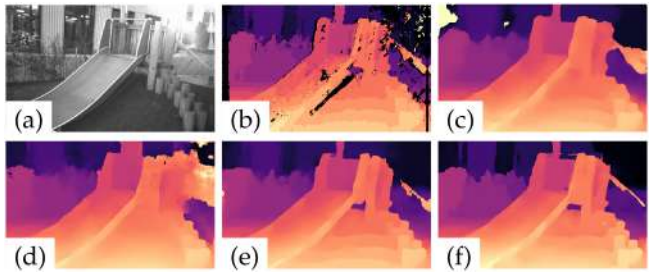

Fig. 5. *Qualitative Results – Data Augmentation.* We report the reference image of a stereo pair from ETH3D [93] (a), the corresponding disparity map from SGM [14] (b) and the predicted disparities using our network trained without color and depth (c), without color (d), without depth (e) and, finally, with our full data augmentation (f).

fact that the late fusion approach enables more tailored feature extraction for the RGB image and the disparity map, rather than relying on a single feature extractor to handle both types of data.

Number of Neurons. Table III presents an ablation study investigating a different number of neurons in MLP_c and MLP_o. Starting from the original configuration [8] (dubbed *Normal*) described in Section IV-A, we design three variants, namely *Tiny*, *Small*, and *Large*, characterized by different numbers of neurons, i.e.reduced to 1/4 for *Tiny*, halved for *Small*, and doubled for *Large*. Results showcase that *Tiny* configuration slightly degrades performance metrics compared to *Normal*, while the *Small* version often outperforms *Tiny* but tends to hover slightly below the *Normal* setting. Conversely, the *Large* version achieves comparable or slightly improved metrics at the expense of increased computational load. In conclusion, reducing neurons below default slightly affects results but speeds up inference. Doubling neurons yields similar outcomes but escalates computational demands. *Normal* strikes a balance between accuracy and computational efficiency across methods.

Data Augmentation. In Table IV, we measure the impact of the different data augmentation strategies during training – specifically, random depth rescaling (Depth Aug.) and color transformations (Color Aug.) – as well as the impact of the batch size. In this experiment, we also report results on Middlebury v3, KITTI 2015 and ETH3D to better appreciate the effect of data augmentation on the zero-shot cross generalization task. Any experiment is conducted by processing disparity maps estimated by SGM [14] at testing time. Observing the table, we can see how enabling Depth Aug. alone slightly reduces the EPE on the SceneFlow test set, while both SEE and bad2 metrics increase. On Middlebury, we observe the opposite behavior, with EPE increasing and bad2 decreasing, while on KITTI and ETH3D, the augmentation always yields lower errors, dramatically lower in terms of EPE on ETH3D. We attribute this to the very different disparity distribution characterizing the ETH3D dataset, which is never observed during training without the considered augmentation. Running Color Aug. alone yields worse results on SceneFlow, but greatly improves the results on the three real-world datasets, notably alleviating the synthetic-to-real domain shift. Not surprisingly, the combination of the two data augmentations yields further improvements on real-world datasets, at the expense of slightly lower performance on SceneFlow. Fig. 5 qualitatively shows the importance of the considered augmentations on the ETH3D test set. Finally, we demonstrate the impact of a larger batch size B (32, the maximum allowed on a single 3090 GPU utilizing the VGG-13 backbone) in significantly enhancing results on both synthetic and real data. These findings provide strong evidence that leveraging appropriate data augmentation techniques alongside a larger batch size can be particularly advantageous in tackling the challenge of the synthetic-to-real domain shift.

Encoder Backbone. We conclude our ablation study by evaluating several variants of our Neural Disparity Refinement network, each characterized by a different backbone used as a feature extractor for both the RGB image and the noisy disparity map. Table V presents results from various backbones on the SceneFlow test set, including parameter counts and runtime analysis. This analysis covers total training time, feature extraction time (referring to the time required by the specific backbone to extract features), overall inference time (which remains constant at approximately 57 ms in our experiments for MLPs inference), and the corresponding FPS averaged across the entire test set. It is important to note that feature extraction time, inference time, and FPS all refer to the inference process on full-resolution images of 960×540 pixels. Each model has been trained with the largest batch size suitable for running on a single 3090 GPU. This approach ensures that each neural network can fully utilize the GPU's memory capacity, maximizing computational efficiency and achieving optimal training performance for every backbone. By avoiding a uniform batch size, which could hinder some models from utilizing the GPU's resources effectively, we guarantee that each architecture is thoroughly exploited, resulting in superior outcomes across our experiments. Additionally, we have matched the number of training epochs for each backbone to the corresponding number of iterations reported in [8]. As a reference to our prior work [8], we also report our original results using a VGG-13 encoder and training with batch 8. We can observe how the latter, trained with batch 32, results in competitive and often better performance compared to more modern backbones on some metrics, e.g.,

TABLE IV
ABLATION STUDY - DATA AUGMENTATION

Depth Aug.	Color Aug.	Batch Size	SceneFlow [3]			Midd-T(Q) [92]		K-15 [90]		ETH3D [93]	
			EPE	SEE	bad2	EPE	bad2	EPE	bad3	EPE	bad1
-	-	-	9.51	8.51	23.34	4.22	17.7	6.43	14.0	1.43	13.4
-	-	8	1.08	1.14	5.27	2.84	10.71	3.26	8.32	15.66	14.71
✓	-	8	1.05	1.17	5.40	3.33	10.36	2.35	7.04	2.67	14.06
-	✓	8	1.13	1.30	5.18	1.19	9.17	1.95	5.96	0.62	5.04
✓	✓	8	1.19	1.26	5.80	1.04	7.93	**1.60**	**5.41**	0.44	5.20
✓	✓	32	**1.04**	**1.04**	**4.60**	**1.03**	**7.80**	1.67	5.43	**0.43**	**4.78**

We report the results obtained by our network when trained with different augmentation strategies. During the evaluation, our network refines the disparity maps estimated by SGM [14]. The bold values indicate the best results.

TABLE V
ABLATION STUDY - ENCODER BACKBONE

Input	Backbone	Batch	Epochs	EPE	SEE	bad2	bad3	bad4	bad5	∼ Time Train (h)	Back. Feat. (ms)	Tot. (ms)	FPS	Size (Mb)
	Initial Depth	-	-	24.68	21.78	46.23	45.79	45.46	45.20	-	-	-	-	-
AD-Census [87]	VGG-13 [8], [85]	8	100	1.53	1.48	8.49	6.10	4.86	4.10	23	16	73	13.70	22.88
	VGG-13 [85]	32	400	1.26	1.20	6.30	4.57	3.68	3.13	53	16	73	13.70	22.88
	ResNet-18 [86]	32	400	1.29	1.23	6.72	4.85	3.89	3.30	48	14	71	14.08	33.64
	WideResNet-50 [94]	24	300	**1.17**	1.15	5.82	4.30	3.52	3.03	75	27	84	11.90	77.97
	ConvNeXt [95]	12	150	1.22	1.17	6.38	4.70	3.83	3.28	232	62	119	8.40	190.93
	RegNetX-16GF [96]	32	400	1.29	1.29	6.40	4.54	3.61	3.06	87	24	81	12.34	30.29
	ResNeXt-50 ($32 \times 4d$) [97]	24	300	**1.02**	**1.01**	**5.25**	**3.85**	**3.13**	**2.69**	75	27	84	11.90	78.04
	MpViT-S [98]	12	150	1.22	**1.07**	**5.09**	**3.52**	**2.75**	**2.33**	91	75	132	7.58	55.36
	CoaT-Lite-S [99]	12	150	**0.99**	1.12	**5.39**	**3.87**	**3.12**	**2.66**	80	67	124	8.06	49.51
	PoolFormer-S12 [100]	12	150	1.23	**1.11**	6.19	4.61	3.79	3.29	43	28	85	11.76	35.34
	Initial Depth	-	-	13.46	10.84	27.09	24.86	23.67	22.86	-	-	-	-	-
C-CNN [88]	VGG-13 [8], [85]	8	100	1.42	1.36	7.11	5.25	4.28	3.68	23	16	73	13.70	22.88
	VGG-13 [85]	32	400	1.31	1.19	6.75	4.93	4.02	3.46	53	16	73	13.70	22.88
	ResNet-18 [86]	32	400	1.34	1.23	5.92	4.35	3.57	3.08	48	14	71	14.08	33.64
	WideResNet-50 [94]	24	300	1.28	1.21	6.02	4.42	3.64	3.16	75	27	84	11.90	77.97
	ConvNeXt [95]	12	150	1.35	1.31	6.67	4.93	4.03	3.49	232	62	119	8.40	190.93
	RegNetX-16GF [96]	32	400	1.33	1.34	6.29	4.42	3.54	3.05	87	24	81	12.34	30.29
	ResNeXt-50 ($32 \times 4d$) [97]	24	300	**1.11**	**1.12**	**5.31**	**3.86**	**3.13**	**2.68**	75	27	84	11.90	78.04
	MpViT-S [98]	12	150	**1.26**	**1.17**	**5.50**	**3.68**	**2.83**	**2.35**	91	75	132	7.58	55.36
	CoaT-Lite-S [99]	12	150	**1.06**	**1.16**	**5.60**	**3.99**	**3.20**	**2.72**	80	67	124	8.06	49.51
	PoolFormer-S12 [100]	12	150	1.37	1.24	6.51	4.89	4.09	3.59	43	28	85	11.76	35.34
	Initial Depth	-	-	9.51	8.51	23.34	21.49	20.52	19.88	-	-	-	-	-
SGM [14]	VGG-13 [8], [85]	8	100	1.19	1.26	5.80	4.33	3.56	3.07	23	16	73	13.70	22.88
	VGG-13 [85]	32	400	**1.04**	1.04	4.60	3.47	2.87	2.50	53	16	73	13.70	22.88
	ResNet-18 [86]	32	400	1.10	1.13	5.29	3.91	3.19	2.73	48	14	71	14.08	33.64
	WideResNet-50 [94]	24	300	**1.04**	1.04	4.84	3.62	2.99	2.59	75	27	84	11.90	77.97
	ConvNeXt [95]	12	150	1.09	1.11	5.28	4.01	3.35	2.93	232	62	119	8.40	190.93
	RegNetX-16GF [96]	32	400	1.12	1.11	5.25	3.74	3.02	2.60	87	24	81	12.34	30.29
	ResNeXt-50 ($32 \times 4d$) [97]	24	300	**0.90**	**0.96**	**4.25**	**3.14**	**2.58**	**2.23**	75	27	84	11.90	78.04
	MpViT-S [98]	12	150	1.12	**0.98**	**4.41**	**3.05**	**2.38**	**1.99**	91	75	132	7.58	55.36
	CoaT-Lite-S [99]	12	150	**0.85**	**1.01**	**4.36**	**3.17**	**2.56**	**2.20**	80	67	124	8.06	49.51
	PoolFormer-S12 [100]	12	150	**1.04**	**0.98**	4.86	3.69	3.09	2.71	43	28	85	11.76	35.34

We report the results obtained by our network when deploying a variety of different backbones, either based on CNNs, Transformers or both. We highlight *first* , *second* and *third* absolute bests. "Train (h)" refers to total training time, "Back. Feat. (ms)" to feature extraction time for each backbone, while "Tot. (ms)" and "FPS" represent overall inference time and frames per second, respectively.
The bold values indicate the best results.

ResNet-18, ResNetX-16GF, and PoolFormer-S12, but is consistently outperformed by others such as ResNeXt-50, MpViT-S, and CoaT-Lite-S, the latter two being based on modern Transformers [98], [99]. This outcome suggests that the Neural Disparity Refinement network can benefit from recent advances in network architecture. However, the specific choice of backbone may depend on the desired trade-off between accuracy and other factors such as the number of parameters and running time. It is worth noting that all the backbones have been trained from scratch without any pre-training on additional tasks or datasets (e.g., ImageNet).

D. Stereo Refinement

We now conduct a set of experiments to assess the performance of our framework in refining disparities produced by standard stereo algorithms or deep stereo networks. In the remaining experiments, we also report results achieved by the three best variants of our framework according to results in Table V, i.e., using ResNeXt-50, MpViT-S and CoaT-Lite-S as the backbones for the feature encoders.

Comparison With Refinement Methods. Here, we aim to compare the performance of our Neural Disparity Refinement network to other existing methods for refining disparity maps. To this end, we consider two state-of-the-art approaches: DRR [10] and RecResNet [11]. We evaluate the performance of these methods on the Middlebury v3 dataset, using disparity maps produced by the C-CNN algorithm (a), and on the KITTI 2015 dataset, using the output of C-CNN (b) or the OpenCV SGBM algorithm (c). The results of these comparisons are presented in Table VI. In our previous studies, our Neural Disparity Refinement framework [8] already outperformed other competing methods. Nonetheless, in the new experiments, we find that it performs even better when used with advanced backbones such as ResNeXt-50, MpViT-S, and CoaT-Lite-S, which incorporate recent innovations in network design and have demonstrated superior performance in other tasks.

TABLE VI
COMPARISON WITH REFINEMENT FRAMEWORKS

Midd-T [92]	bad2		EPE	
Method	Non-Occ	All	Non-Occ	All
C-CNN [88]	18.24	26.71	6.06	8.71
+ DRR [10]	12.85	17.83	1.77	2.37
+ DRR ×2 [10]	11.53	16.41	1.79	2.32
+ **Ours** (VGG) [8]	11.22	15.30	1.44	1.88
+ **Ours** (ResNeXt-50)	**10.09**	**13.65**	**1.32**	**1.65**

(a)

K-15 [90]	D1		EPE	
Method	Non-Occ	All	Non-Occ	All
C-CNN [88]	6.41	8.25	1.70	2.46
+ DRR ×2 [10]	2.58	3.08	0.78	0.84
+ RecResNet [11]	-	3.46	-	-
+ **Ours** (VGG) [101]	2.27	2.60	0.75	0.79
+ **Ours** (ResNeXt-50)	**1.97**	**2.24**	**0.74**	**0.78**

(b)

K-15 [90]	D1
Method	All
SGBM [14]	5.15
+ Dil-Net (ref.) [12]	4.58
+ Dil-Net (fus.) [12]	3.07
+ **Ours** (VGG) [8]	2.93
+ **Ours** (ResNeXt-50)	**2.21**

(c)

In (a), all models are trained on the SceneFlow dataset [3] and tested on the 15 images of the Middlebury v3 [92] training dataset at quarter resolution. In (b) and (c), models are fine-tuned on the first 160 images of the KITTI 2015 [90] training set and evaluated on the remaining 40.
The bold values indicate the best results.

TABLE VII
END-TO-END NETWORKS AS STEREO BLACKBOX

Input	EPE	SEE	bad2
GANet [40]	0.95	1.83	4.81
GANet [40] + Ours (VGG-13 [85]) [8]	0.96	1.43	4.78
GANet [40] + Ours (ResNeXt-50 [97])	**0.89**	**1.31**	**4.45**
GANet [40] + Ours (MpViT-S [98])	1.23	1.51	4.98
GANet [40] + Ours (CoaT-Lite-S [99])	0.91	1.47	4.74
AANet [102]	1.10	2.81	5.84
AANet [102] + Ours (VGG-13 [85]) [8]	1.15	1.62	5.43
AANet [102] + Ours (ResNeXt-50 [97])	**1.08**	**1.49**	**5.10**
AANet [102] + Ours (MpViT-S [98])	1.32	1.71	6.52
AANet [102] + Ours (CoaT-Lite-S [99])	1.09	1.69	5.39
HSMNet [44]	1.86	3.77	11.47
HSMNet [44] + Ours (VGG-13 [85]) [8]	1.54	1.86	9.14
HSMNet [44] + Ours (ResNeXt-50 [97])	**1.29**	**1.59**	**8.01**
HSMNet [44] + Ours (MpViT-S [98])	1.56	1.65	9.23
HSMNet [44] + Ours (CoaT-Lite-S [99])	1.48	1.80	9.12
RAFT-Stereo [5]	**0.67**	4.17	4.31
RAFT-Stereo [5] + Ours (VGG-13 [85]) [8]	0.76	1.18	3.84
RAFT-Stereo [5] + Ours (ResNeXt-50 [97])	0.79	1.53	3.76
RAFT-Stereo [5] + Ours (MpViT-S [98])	1.12	1.48	2.85
RAFT-Stereo [5] + Ours (CoaT-Lite-S [99])	0.71	**1.17**	**2.72**

We validate our model using disparity maps computed by endto- end stereo networks, using the official weights released by the authors after training on the SceneFlow dataset [3].
The bold values indicate the best results.

Refinement on Deep Stereo Models. As previously mentioned, our framework is designed to refine the output of any stereo algorithm or deep stereo network. To demonstrate its effectiveness at improving the output of state-of-the-art deep stereo architectures, we conducted experiments using four leading methods: GANet [40], AANet [102], HSMNet [36], and the more recent RAFT-Stereo [5] network. The results, obtained from evaluations on the SceneFlow test set and summarized in Table VII, consistently highlight our network's capability to enhance the accuracy of already high-quality disparity maps produced by these end-to-end models when used with various backbones. However, notably, when combined with ResNeXt-50 as the backbone, our approach demonstrates a consistently superior performance compared to other configurations. This indicates that our method, particularly in conjunction with ResNeXt-50, exhibits the most promising results in refining disparity maps and effectively mitigating bleeding artifacts present in outputs generated by these end-to-end networks. This ability can be observed by looking at the SEE improvement in Table VII, and qualitatively on the point clouds shown in Fig. 6. These facts demonstrate the robustness and versatility of our approach, as it can effectively refine the output of a variety of stereo algorithms and networks.

TABLE VIII
GENERALIZATION PERFORMANCE

Method	Target Domain: K-15 [90]	Midd-T [92] F	Midd-T [92] H	Midd-T [92] Q	ETH3D [93]
SGM [14]	15.35	31.19	20.34	15.56	15.24
Ours (VGG-13) [8], [85]	5.41	27.87	17.70	11.75	5.20
Ours (VGG-13) [85]	5.43	**25.95**	**15.91**	**10.97**	4.78
Ours (ResNet18) [86]	5.17	26.22	16.24	12.41	5.13
Ours (WideResNet-50) [94]	5.26	**25.38**	**15.79**	**11.06**	**4.76**
Ours (ResNeXt-50) [97]	**5.00**	**25.29**	**15.35**	**10.59**	**4.25**
Ours (RegNetX-16GF) [96]	**5.11**	29.86	18.03	12.01	7.44
Ours (CoaT-Lite-S) [99]	5.92	29.15	17.32	13.21	5.80
Ours (ConvNeXt) [95]	**5.11**	28.46	19.16	11.84	5.26
Ours (MpViT-S) [98]	**4.98**	29.58	16.60	11.24	50.09
Ours (PoolFormer-S12) [100]	5.12	26.57	16.96	12.24	**4.71**

(a)

SGM [14]	15.35	31.19	20.34	15.56	15.24
SGM [14] + Ours	**5.00**	**25.29**	**15.35**	**10.59**	**4.25**
AD-Census [87]	22.28	42.68	28.81	23.21	21.27
AD-Census [87] + Ours	**5.36**	**37.06**	**20.11**	**11.77**	**8.27**
PatchMatch [103]	37.06	55.15	42.97	36.60	47.80
PatchMatch [103] + Ours	**9.19**	**45.89**	**29.01**	**16.57**	**23.78**

(b)

PSMNet [38]	7.86	33.69	21.69	17.24	23.19
GANet [40]	10.46	45.36	26.75	15.52	8.68
DSMNet [61]	5.50	29.95	**16.88**	13.75	12.52
CFNet [47]	6.01	29.12	20.11	13.77	5.77
ITSA-CFNet [66]	**4.96**	26.38	18.01	**13.32**	**5.40**
Graft-PSMNet [65]	**5.34**	**25.46**	17.81	14.18	11.42
RAFT-Stereo [5]	5.74	**18.33**	**12.59**	**9.36**	**3.28**
SGM [14] + Ours	**5.00**	**25.29**	**15.35**	**10.59**	**4.25**

(c)

RAFT-Stereo [5]	5.74	18.33	12.59	9.36	3.28
RAFT-Stereo [5] + Ours	**5.19**	**17.53**	**11.51**	**9.28**	**2.73**

(d)

All methods are trained on SceneFlow and tested on KITTI [90], Middlebury [92], and ETH3D [93]. Errors are the percentage of pixels with EPE greater than the specified threshold: 3px for KITTI, 2px for Middlebury, 1px for ETH3D. Every method is evaluated using the authors' code, on All pixels. We highlight *first* , *second* and *third* bests in single sub-tables.
The bold values indicate the best results.

Zero-Shot Generalization. To support the robustness of our framework, we compare the accuracy it achieves when trained on synthetic data and transferred to real data without any fine-tuning. Table VIII collects several experiments aimed at thoroughly assessing this aspect, by evaluating the accuracy achieved by models trained on the SceneFlow dataset alone and tested on KITTI 2015, Middlebury v3, and ETH3D. On top (a), we first report the results achieved by our framework with different backbones when processing the disparity maps computed from the SGM [14] algorithm as input. We can notice

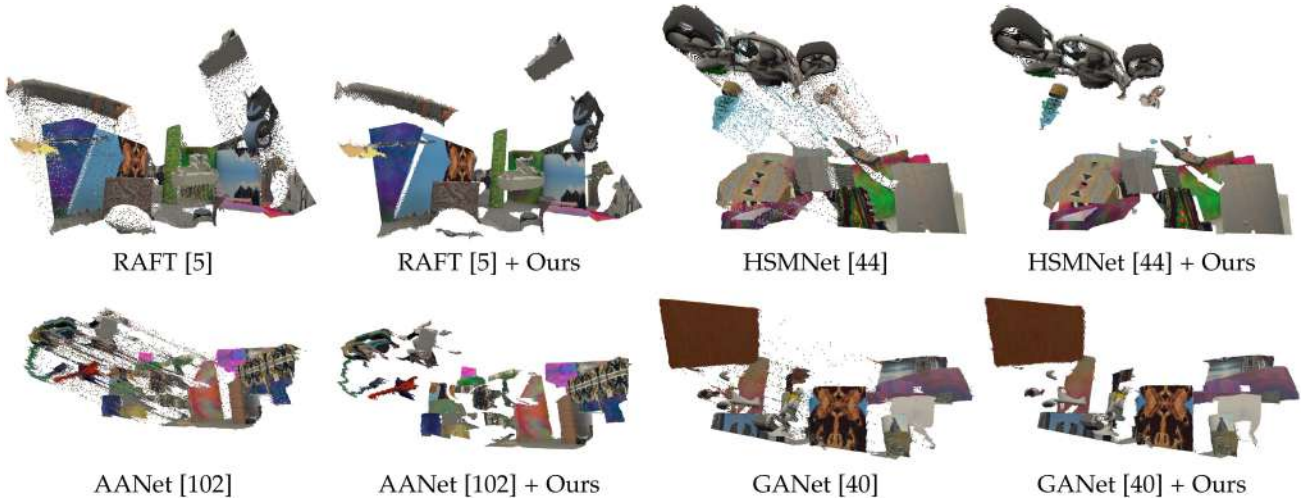

Fig. 6. *Point Cloud Comparison on the SceneFlow dataset*. We show the outcome of different deep stereo networks and the point clouds obtained using our refinement method on the initial disparity estimates. Please zoom in for details.

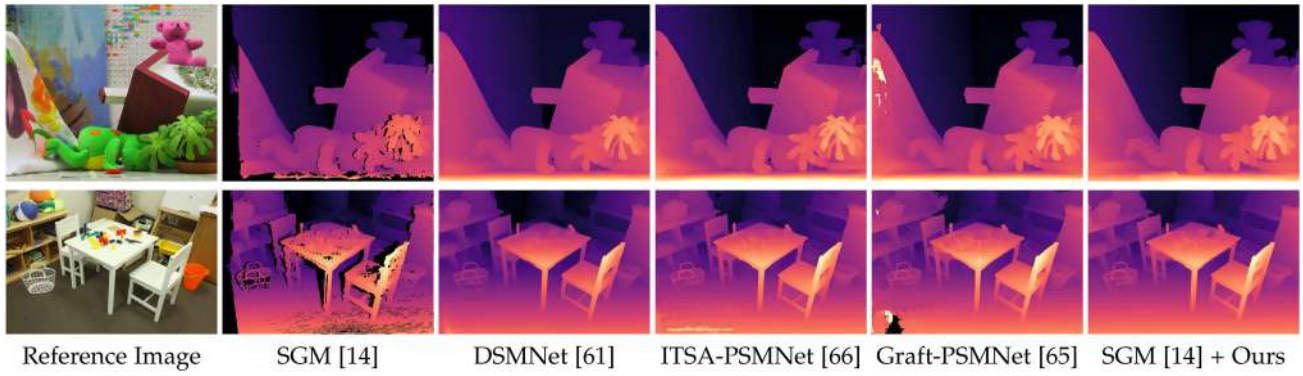

Fig. 7. *Qualitative comparison on the Middlebury v3 dataset* . From left to right: reference image at half resolution, disparity maps computed by the SGM algorithm [14], three state-of-the-art networks designed for zero-cross generalization purposes [61], [65], [66] and, finally, our method applied to the SGM algorithm using ResNeXt-50 as feature extractor.

how our original framework [8] still achieves remarkable results among the many alternatives, with the ResNeXt-50 model being the most effective. From the remaining experiments, we will select this latter as our final model. As a second experiment, in (b) we test our model on the disparity maps produced by the noisier stereo algorithms, such as AD-Census [87] and PatchMatch Stereo [103]. While the performance is not comparable to the one achieved with SGM as input, our framework still significantly improves the initial disparity maps. In (c), we compare with existing state-of-the-art solutions designed purposely to achieve generalization. We can notice how our original framework [8] from (a) already outperforms most of the existing networks, including those specifically designed to achieve robustness to the synthetic to real shift – i.e., ITSA-CFNet [66] and Graft-PSMNet [65]. Our new variant built on the ResNeXT-50 backbone consistently ranks 2nd on the benchmark, with RAFT-Stereo being the only architecture resulting in higher robustness than ours, except on KITTI. Nonetheless, in (d), we demonstrate how our framework can refine the disparity maps predicted by RAFT-Stereo and further improve its generalization performance. A qualitative comparison between these methodologies is visible in Fig. 7.

Training With More Synthetic Data. To further understand the impact of training data on the generalization abilities of our model, we considered the best model from Table VIII, which was built on top of the ResNeXt-50 feature extractor. We used the SceneFlow, UnrealStereo4K-Q, VirtualKITTI2, and CREStereo datasets for training and evaluated the effect of adding additional datasets while maintaining the same number of training iterations. Table IX reports the outcome of this experiment. As shown in the table, including additional synthetic data during training significantly improves the generalization capabilities of our model on the KITTI and Middlebury-Q testing datasets. This behaviour is particularly evident when using stereo pairs from the Virtual KITTI 2 dataset, which features similar autonomous driving environments of KITTI, for training. Nonetheless, the ETH3D dataset and images at the highest resolution (F) of the Middlebury dataset exhibit an exception to this trend, with the best results obtained using the SceneFlow dataset alone. These findings suggest that the zero-shot generalization abilities of our network can be potentially enhanced by combining multiple datasets during training.

Online KITTI Benchmark. To conclude our experiments, we present the updated results of our Neural Disparity Refinement

TABLE IX
ABLATION STUDY - SYNTHETIC TRAINING DATASETS

SceneFlow [3]	UnrealStereo4K (Q) [7]	Virtual KITTI2 [89]	CREStereo (H) [6]	K-15 [3]		Midd-T [92]			ETH3D [93]
				2012	2015	F	H	Q	
✓	-	-	-	5.7	5.0	**25.3**	15.3	10.6	**4.3**
✓	✓	-	-	5.3	4.7	25.4	15.2	10.3	5.7
✓	✓	✓	-	4.9	4.2	26.0	15.0	10.0	6.3
✓	✓	✓	✓	**4.6**	**3.8**	26.2	**14.7**	**9.4**	5.4

We report results achieved by our framework (with a ResNeXt-50 backbone [86]) when trained on multiple synthetic datasets. During evaluation, we use the SGM [14] algorithm to compute disparity maps, which are then refined by our framework.
The bold values indicate the best results.

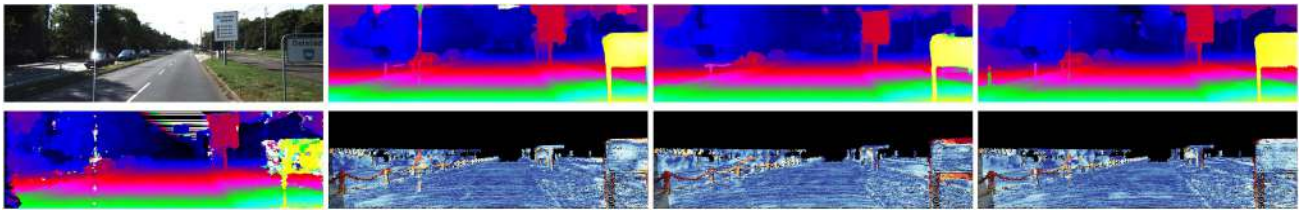

Fig. 8. *Qualitative comparison on the KITTI 2015 benchmark.* From left to right: reference image and noisy input disparity from [88], disparity (top) and corresponding error maps (bottom) for DRR [10], RecResNet [11], and our method.

TABLE X
EVALUATION ON THE KITTI 2015 BENCHMARK

Method	**D1-all**	D1-fg	D1-bg
Disparity Refinement			
Dil-Net [12]	3.92	7.44	3.22
DRR $\times 2$ [10] †	3.16	6.04	2.58
LRCR [104]	3.03	5.42	2.55
RecResNet [11]	3.10	6.30	2.46
Ours (VGG) [101] †	2.35	3.93	2.03
Ours (ResNeXt-50) †	**2.21**	**3.87**	**1.88**
End-to-End Stereo			
AANet [102]	2.55	5.39	1.99
PSMNet [38]	2.32	4.62	1.86
HSMNet [44]	2.14	3.85	1.80
HITNet [37]	1.98	3.20	1.74
GANet [40]	1.81	3.46	**1.48**
DSMNet [61]	**1.77**	**3.23**	**1.48**

Methods indicated with † consider the disparity maps computed by C-CNN [88] as noisy input of the network. The other refinement methods adopt different noisy disparity inputs as described in their papers.
The bold values indicate the best results.

TABLE XI
RESULTS WITHOUT AND WITH MLP_{conf}

	KITTI [90]		Midd-T [92]			ETH3D [93]
	2012	2015	F	H	Q	
w/o MLP_{conf}	**4.6**	**3.8**	26.2	**14.7**	9.4	5.4
with MLP_{conf}	**4.6**	4.1	**25.4**	**14.7**	**9.1**	**5.1**

Results achieved by our framework (ResNeXt-50 backbone [86]) when trained on multiple synthetic datasets, without or with $\text{MLP}conf$, refining SGM [14] disparity maps.
The bold values indicate the best results.

network on the KITTI 2015 online benchmark. Specifically, we evaluate the ability of our network to refine raw disparity maps estimated by the C-CNN algorithm [88] to be comparable with other frameworks. To do this, we select the ResNeXt-50 variant of our network and fine-tune it on the KITTI 2015 training set for 500 epochs. The results, reported in Table X, show that our re-implemented framework outperforms our previously published results [8], achieving better performance compared to other deep-learning-based refinement strategies, as well as comparable performance to state-of-the-art end-to-end stereo architectures. Qualitative comparisons to these methodologies on the KITTI benchmark are visible in Fig. 8.

E. More Tasks

To demonstrate the versatility and effectiveness of our framework, we put it to the test on three additional tasks: unbalanced stereo matching, confidence estimation, and multi-view stereo refinement. The results of these experiments show that our approach can successfully tackle a wide range of depth estimation problems.

1) Confidence Estimation: While primarily focused on refining disparity maps, our framework exhibits flexibility to extend its functionality by estimating a confidence for the input disparity. This dual capability allows us to reinforce the network in two key areas. First, it results in slight enhancements in disparity accuracy across diverse datasets, as demonstrated in Table XI. This improvement primarily arise from the concurrent nature of joint training, where both estimating confidence and refining the disparity map occur simultaneously. This interconnected relationship is made possible by utilizing two distinct loss functions throughout the network's training process: one specifically targets the refinement of the disparity map, while the other concentrates on estimating confidence. Consequently, our model demonstrates an interconnected learning process, wherein confidence estimation and refined map generation mutually influence each other, promoting a more robust learning mechanism.

Second, and of greater importance, despite the minor advancements in disparity accuracy, the ability to simultaneously handle both tasks—refining disparity maps and estimating confidence measures—constitutes a crucial aspect of our framework.

We demonstrate the effectiveness of this confidence measure in identifying outliers in the input disparity maps. More specifically, we measure the Area Under the Curve (AUC) as defined in [9], [27], [69]: pixels of the disparity map are sorted in decreasing order of confidence, and the badτ metric is gradually calculated on sparse maps obtained by iteratively sampling (e.g.,

TABLE XII
EXPERIMENTAL RESULTS – CONFIDENCE ON THE INPUT DISPARITY

Census-CBCA	K-12	K-15	Midd-T (Q)	ETH3D
ENS_{23}	6.62	5.60	11.15	8.20
GCP	6.37	5.29	11.18	8.54
LEV_{22}	5.75	4.56	11.47	8.33
LEV_{50}	5.67	4.49	11.88	8.70
FA	6.01	4.71	13.34	12.68
ENS_7	7.53	6.28	14.59	10.92
O1	6.15	4.77	11.45	9.73
O2	5.81	4.53	11.06	9.28
CCNN	5.76	4.40	11.24	9.09
$PBCP_r$	6.01	4.89	10.20	9.19
$PBCP_d$	5.54	4.44	15.01	14.43
EFN	6.16	4.74	13.64	9.84
LFN	5.80	4.43	11.81	9.02
MMC	5.71	4.36	11.26	8.98
ConfNet	6.11	4.85	11.85	8.24
LGC	5.59	4.25	9.93	7.58
Ours*	5.42	**4.19**	**7.69**	**5.81**
RCN	14.79	13.29	17.59	13.21
MPN	5.58	4.31	9.00	6.23
UCN	5.52	4.28	9.17	6.55
LAF	5.33	4.20	10.30	9.50
ACN	5.69	4.35	8.86	6.49
CRNN	11.81	10.48	15.52	11.62
CVA	8.12	6.39	11.04	9.63
Opt.	4.72	3.40	5.31	4.07
D1(%)	27.19	22.28	28.70	21.27

MCCNN-CBCA	K-12	K-15	Midd-T (Q)	ETH3D
ENS_{23}	3.53	3.76	9.58	14.48
GCP	3.44	3.44	9.86	15.69
LEV_{22}	3.05	3.05	8.48	13.74
LEV_{50}	2.89	2.97	8.45	13.45
FA	3.00	3.02	8.33	14.00
ENS_7	3.94	4.33	10.95	16.37
O1	2.96	2.93	8.25	15.18
O2	2.79	2.87	8.09	14.92
CCNN	2.84	2.91	8.23	14.23
$PBCP_r$	3.27	3.48	7.86	14.98
$PBCP_d$	2.95	3.06	11.26	17.10
EFN	3.22	3.15	9.44	13.98
LFN	3.04	3.00	8.45	14.15
MMC	2.95	2.94	8.15	13.78
ConfNet	3.22	3.32	8.31	13.34
LGC	2.96	2.77	8.09	14.37
Ours*	2.91	2.90	**6.98**	**12.81**
RCN	5.53	5.35	17.75	24.99
MPN	3.03	3.09	8.26	13.02
UCN	2.85	2.90	8.07	13.01
LAF	2.81	2.90	8.03	13.17
ACN	2.94	3.03	8.24	13.09
CRNN	5.08	4.87	17.75	25.02
CVA	3.31	3.38	9.34	15.78
Opt.	2.35	2.16	5.63	10.31
D1(%)	18.88	16.93	29.80	34.28

Census-SGM	K-12	K-15	Midd-T (Q)	ETH3D
ENS_{23}	1.99	2.18	12.82	7.82
GCP	2.02	2.47	13.12	6.34
LEV_{22}	1.79	1.98	12.20	6.79
LEV_{50}	1.72	1.89	12.50	7.47
FA	2.24	2.35	13.02	7.55
ENS_7	2.71	2.86	15.21	8.71
O1	1.69	1.72	11.40	7.20
O2	1.64	1.55	10.82	8.08
CCNN	1.90	1.92	12.01	7.39
$PBCP_r$	1.87	2.03	10.73	7.33
$PBCP_d$	1.92	2.04	16.27	14.03
EFN	2.27	2.14	13.98	7.53
LFN	2.02	2.04	12.74	8.04
MMC	1.75	1.69	11.64	7.89
ConfNet	2.33	2.27	13.84	7.08
LGC	1.80	1.49	10.89	6.63
Ours*	**1.16**	**1.24**	**7.57**	**4.06**
RCN	3.04	2.65	20.67	11.94
MPN	1.57	1.67	8.92	5.16
UCN	1.57	1.62	8.88	5.28
LAF	1.44	1.60	10.44	6.44
ACN	1.70	1.74	9.20	5.19
CRNN	3.14	2.80	18.54	10.49
CVA	2.20	2.23	10.65	6.45
SGMF	2.16	2.28	11.04	5.47
Opt.	0.79	0.74	4.57	2.14
D1(%)	10.33	9.00	26.68	15.74

MCCNN-SGM	2012	2015	Midd.	ETH
ENS_{23}	0.82	1.64	7.94	5.39
GCP	1.00	1.91	7.53	5.61
LEV_{22}	0.81	1.37	8.43	4.13
LEV_{50}	0.75	1.15	6.46	3.92
FA	1.08	1.33	7.76	5.50
ENS_7	1.27	1.82	9.84	6.55
O1	0.80	1.21	6.46	5.03
O2	0.72	1.07	6.24	5.36
CCNN	0.89	1.22	7.64	5.11
$PBCP_r$	0.86	1.25	6.09	5.07
$PBCP_d$	1.08	1.44	12.71	12.18
EFN	1.27	1.41	9.52	4.71
LFN	0.99	1.17	7.75	5.35
MMC	0.93	1.11	7.01	4.75
ConfNet	0.87	1.36	6.93	3.40
LGC	0.85	1.10	6.87	4.83
Ours*	**0.51**	**0.85**	**4.83**	**3.26**
RCN	1.02	2.52	22.54	12.38
MPN	0.63	1.14	6.76	3.82
UCN	0.67	1.19	6.32	3.54
LAF	0.61	1.21	6.01	3.72
ACN	0.63	1.22	7.31	3.83
CRNN	0.98	2.24	21.80	12.20
CVA	0.77	1.49	8.61	6.06
SGMF	0.83	1.83	6.46	4.25
Opt.	0.25	0.44	2.94	1.41
D1(%)	6.08	6.03	21.80	12.59

GANet	2012	2015	Midd.	ETH
ENS_{23}	3.18	4.52	18.28	5.58
GCP	4.19	5.39	21.00	5.24
LEV_{22}	2.26	3.32	17.58	3.70
LEV_{50}	1.99	2.87	17.87	4.17
FA	4.59	5.68	23.09	7.92
ENS_7	5.05	6.00	22.46	6.37
O1	3.20	4.02	22.54	7.36
O2	2.76	3.67	22.03	7.10
CCNN	3.29	3.93	23.34	7.02
$PBCP_r$	4.02	4.93	19.83	8.59
$PBCP_d$	3.46	4.14	22.50	12.56
EFN	4.84	5.36	23.83	4.96
LFN	3.77	4.25	23.00	6.13
MMC	3.84	4.49	23.89	5.45
ConfNet	5.46	5.21	22.31	4.69
LGC	3.53	4.61	22.42	6.31
Ours*	1.97	**2.41**	19.93	6.11
RCN	2.81	3.82	16.12	6.04
MPN	3.89	4.40	26.64	12.67
UCN	2.62	3.18	23.31	12.23
LAF	1.70	2.58	18.19	7.40
ACN	2.17	2.96	17.99	6.89
CRNN	2.41	3.20	16.33	6.18
CVA	4.00	4.90	23.12	8.75
Opt.	0.57	0.85	5.28	0.99
D1(%)	8.62	10.02	28.61	10.80

We report results on KITTI 2012 [91], 2015 [90], Middlebury v3 [92] and ETH3D [93] datasets in comparison with state-of-the-art methods for confidence estimation [9] when processing disparity maps produced by five stereo algorithms/networks.
The bold values indicate the best results.

5% of pixels each time) from the dense map. Plotting these values produces a ROC curve, whose AUC quantitatively assesses the effectiveness of the confidence measure at removing outliers (the lower the AUC, the better the measure is at identifying and removing outliers). Table XII shows the results achieved by our framework when processing the disparity maps computed by five standard algorithms that are commonly used to evaluate the effectiveness of confidence measures. Specifically, we use noisy disparities obtained using the Census-CBCA, MCCNN-CBCA, Census-SGM, MCCNN-SGM, and GANet algorithms, as defined in [9]. As a reference, we also report the results achieved by state-of-the-art learning-based approaches. However, it is important to note that these approaches are typically trained on real-world datasets with ground-truth depth acquired from active sensors (such as LiDAR), so their results may not be directly comparable to our protocol. It is also worth noting that each learning method in the table has been trained on the specific stereo algorithm used at test time, resulting in multiple instances of the same network. In contrast, our method has been trained only once on a combination of disparity maps computed by SGM and AD-Census algorithms. Nonetheless, we can see that our framework achieves performance that is close to the optimal values and, in most cases, better results than state-of-the-art deep learning-based approaches, even though our network has been trained on synthetic data only. This outcome demonstrates the effectiveness of using synthetic datasets with pixel-accurate ground-truth for the confidence estimation task and shows that our method does not require expensive active sensors. Fig. 9 reports qualitative results concerning confidence estimated by our framework.

2) Multi-View Stereo Refinement: Here, we demonstrate that the effectiveness of our approach is not limited to stereo disparity maps and can be used to improve the quality of depth maps obtained from other image-based solutions. For example, our network can effectively process noisy depth maps estimated by the popular COLMAP system [13], [105] on multi-view stereo images.

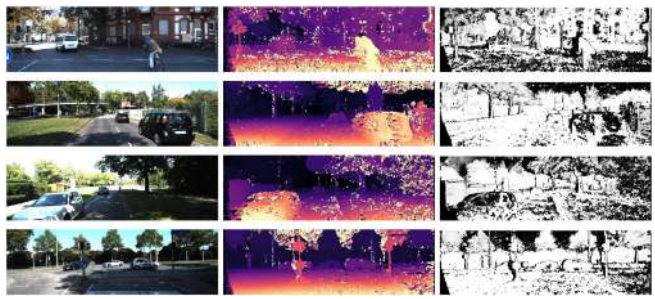

Fig. 9. *Qualitative Results - Confidence Estimation.* From left to right: the reference image, the disparity computed using SGM [14] and the corresponding estimated confidence map using our MLP_{conf} on the KITTI 2015 dataset [90].

To this aim, we pre-process the depth maps generated by COLMAP before feeding our model. This involves applying a series of pre-processing steps, driven by the fact that our neural network is specifically trained to refine disparity maps within the domain of inverse depth, as opposed to directly operating on the depth domain. First, the depth maps are converted into an inverse depth representation, denoted as $\mathcal{D}_{\text{inv}}$, by taking the reciprocal of the depth values. As a few depth values may result in infinities in the inverse depth map, we address this issue by setting those infinite values to zero. Next, we normalize the inverse depth map $\mathcal{D}_{\text{inv}}$ to a common range. This is achieved through min-max scaling, where the minimum and maximum values, denoted as $\min$ and $\max$, respectively, are obtained from $\mathcal{D}_{\text{inv}}$. The normalization process guarantees that the depth values are now within the interval [0,1]. To ensure compatibility with our deep neural network, which can handle a maximum disparity value represented as $d\max$, we further rescale the normalized inverse depth map by $d\max$. This step prepares the depth maps for effective disparity refinement. Then, our network takes the pre-processed inverse depth map $\mathcal{D}_{\text{inv}}$ as input and predicts the refined inverse depth map. Finally, to obtain the final depth map, we reverse the scaling process by dividing the refined

TABLE XIII
EXPERIMENTAL RESULTS – MULTI-VIEW STEREO ON THE ROBUST MVD BENCHMARK

Approach	KITTI [90]		ScanNet [107]		ETH3D [93]		DTU [108]		T&T [109]		Average	
	rel ↓	δ ↑	rel ↓	δ ↑	rel ↓	δ ↑	rel ↓	δ ↑	rel ↓	δ ↑	rel ↓	δ ↑
COLMAP Dense [13], [105]	26.9	52.7	38.0	22.5	89.8	23.2	20.8	69.3	25.7	**76.4**	40.2	48.8
COLMAP Dense [13], [105] + Ours (ResNeXt-50 [86])	**4.7**	**61.9**	**11.2**	**33.6**	**18.6**	**29.4**	**12.0**	**76.4**	**9.3**	59.8	**11.2**	**52.2**

The bold values indicate the best results.

Fig. 10. **Qualitative results – Multi-View Stereo Refinement.** We report four examples, from top to bottom, two examples from ScanNet and two examples from T&T, showing the RGB image and depth maps estimated by COLMAP [13], [105] and refined by our method.

depth estimate by $d\max$ and scaling the values back to their original range using the minimum and maximum values from the min-max scaling. This procedure ensures that the refined depth map accurately represents the correct scene's depth structure.

To evaluate the performance of our method on these depth maps, we measure the absolute relative error (rel) and the percentage of pixels with a minimum ratio between the estimated and ground-truth depth of less than 1.03 (δ, the higher, the better) on the Robust MVD Benchmark [106] – i.e., on the KITTI [90], ScanNet [107], ETH3D [93], DTU [108], and Tank & Temples (T&T) [109] datasets.

The results of this experiment, shown in Table XIII, demonstrate that our model consistently improves the results achieved by COLMAP on all datasets, except for δ on T&T. This is notable because our model was not specifically trained on noisy depths from COLMAP, highlighting its versatility and ability to be applied to a variety of depth representations and scenarios. Fig. 10 shows some qualitatives of RGB images, noisy depth maps from COLMAP, and their refined counterparts obtained using our model.

3) Unbalanced Stereo: In conclusion, we demonstrate the effectiveness of our proposal in addressing the challenging task of unbalanced stereo setups, a common configuration in modern mobile phones where a high-resolution sensor is paired with lower-resolution cameras. In these conditions, the rectification constraint between the two images must be understood to hold up to a scale factor, meaning that the stereo pair is rectified whenever it is resized to the same arbitrary resolution. We denote the unbalance factor between the two images of the stereo pair as κ. For this experiment, we trained again our network from scratch on a combination of the UnrealStereo4K, SceneFlow, Virtual KITTI 2, and CREStereo datasets at their native resolution using the ResNext-50 backbone and setting $d\max$ to 768.

TABLE XIV
EXPERIMENTAL STUDY OF UNBALANCED SETUP

	$\kappa = 4$		$\kappa = 8$		$\kappa = 12$	
Method	EPE	bad3	EPE	bad3	EPE	bad3
	Traditional Stereo					
SGM [14]	28.39	24.64	29.16	33.91	30.84	48.21
+ Ours (VGG-13) [8], [85]	6.69	18.42	8.05	22.60	9.72	29.74
+ Ours (ResNeXt-50 [86])	**4.53**	**16.21**	**5.44**	**20.76**	**6.48**	**27.48**
	End-to-End Stereo					
RAFT-Stereo [5]	4.93	24.33	7.28	36.12	9.88	48.85
+ Ours (VGG-13) [8], [85]	5.26	**22.78**	6.74	**31.77**	8.76	**43.74**
+ Ours (ResNeXt-50) [86]	**4.70**	23.53	**6.43**	33.98	**8.17**	45.19
HSMNet [44]	6.31	21.66	8.67	27.74	11.98	37.17
+ Ours (VGG-13) [8], [85]	5.57	**19.53**	6.81	**23.19**	9.25	**30.16**
+ Ours (ResNeXt-50) [86]	**4.66**	19.57	**6.24**	24.20	**8.82**	32.53

We rely on the UnrealStereo4K dataset to evaluate different methods in unbalanced setups featuring different κ factors.
The bold values indicate the best results.

To evaluate the performance of our approach in unbalanced stereo setups, we conducted experiments on 200 images of the UnrealStereo4K [7] test set. This strategy allowed us to simulate different unbalance factors, κ, and evaluate the accuracy of the estimated disparities using ground-truth information. For this experiment, we use disparity maps computed using both traditional algorithms, such as SGM [14], and two state-of-the-art deep algorithms, RAFT-Stereo [5] and HSMNet [36]. Table XIV summarizes the results of this experiment. For the best performance, for the SGM and RAFT-Stereo algorithms, we downsampled the reference image to match the resolution of the target frame. For the same purpose, we upsampled the target to match the reference image resolution for HSMNet, similar to [8]. The resulting disparity maps are then upsampled to the original, reference image resolution by multiplying them by κ in the case of SGM and RAFT-Stereo. We conducted experiments with unbalance factors of 4, 8, and 12 and refined the disparity maps using both our original VGG-based framework and the newly evaluated ResNeXt-50 variant. Regardless of the backbone, our method consistently improves the accuracy of all three stereo methods for any value of κ. In the case of refining SGM maps, the ResNeXt-50 variant consistently outperforms the original VGG-13 model. Interestingly, when refining the outputs of deep stereo models, the ResNeXt-50 backbone can achieve lower EPE, albeit at the expense of slightly higher bad3 error. However, combining SGM with our ResNeXt-50 model yields the best overall results for any unbalance factor. Fig. 11 shows examples

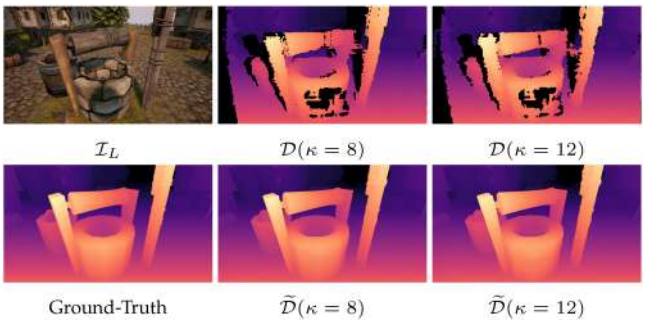

Fig. 11. *Qualitative Results on UnrealStereo4K [7] – Unbalanced Setting.* The top row depicts the input image, $\mathcal{I}_l$, at 3840×2160 and the disparity maps, $\mathcal{D}$, computed by SGM when the right image, $\mathcal{I}_r$, is 480×270 and 320×180 ($\kappa = 8$ and 12). The bottom row shows the ground-truth and estimated disparity $\widetilde{\mathcal{D}}$ at 3840×2160.

TABLE XV
GENERALIZATION TO MIDDLEBURY V3 - UNBALANCED SETUP

		EPE	bad2	bad3
$\kappa = 2$	SGM [14]	37.11	36.54	33.84
	+ Ours (VGG-13) [8], [85]	8.55	28.82	22.71
	+ Ours (ResNeXt-50) [86]	**6.86**	**23.59**	**18.38**
$\kappa = 4$	SGM [14]	28.07	37.69	33.11
	+ Ours (VGG-13) [8], [85]	11.35	26.96	21.77
	+ Ours (ResNeXt-50) [86]	**9.84**	**22.13**	**17.89**

We consider $\kappa = \{2, 4\}$ and evaluate our models trained on SceneFlow (SF) and UnrealStereo4K (US) at F. Initial disparities, $\mathcal{D}$, are computed by SGM at H and Q for $\kappa = 2$ and $\kappa = 4$, respectively.
The bold values indicate the best results.

of refined disparity maps on the UnrealStereo-4 K dataset using different unbalanced factors. As in [8], we further evaluate the generalization capability of our model in the unbalanced setting by conducting experiments on the Middlebury 2014 dataset [1] without any fine-tuning. Specifically, we use SGM to compute initial disparity maps from unbalanced stereo pairs with unbalance factors of 2 and 4, and then use our framework to refine these maps. The results of this experiment are reported in Table XV, which shows that our framework consistently improves the results achieved by SGM and demonstrates its excellent generalization performance. Moreover, the ResNeXt-50 variant of our model outperforms our previous VGG-13-based model in this task. Overall, these results confirm the effectiveness of our approach in handling unbalanced stereo configurations.

V. LIMITATIONS

Our methodology, while exhibiting considerable efficacy in enhancing disparity maps from diverse sources, encounters limitations when addressing certain scenarios. Particularly, when subjected to non-Lambertian surfaces, our approach faces challenges in accurately refining depth estimations. This limitation is prominently observed in experiments conducted on the Booster [110] dataset using SGM or similar stereo matching algorithms.

The errors become pronounced around transparent objects or highly reflective surfaces, such as mirrors, highlighting a significant challenge in our method's depth estimation accuracy for these specific materials. Despite our Neural Disparity Refinement network significantly improves the overall quality of disparity maps, its performance struggles when dealing with such surfaces due to the absence of non-Lambertian objects in the synthetic datasets utilized during the model's training phase.

Fig. 12. *Qualitative examples on the Booster [110] dataset.* From left to right: RGB image, disparity map computed using the SGM algorithm, refined disparity map using our refinement method, and finally, the aleatoric confidence estimated by our network.

These limitations emphasize the importance of accounting for diverse surface materials and scenarios during model training to enhance its robustness across a wider spectrum of real-world conditions. Addressing this limitation would require incorporating diverse material representations and scenarios into the training data to better equip the model for handling non-Lambertian surfaces effectively.

The visual examples provided in Fig. 12 using the Booster dataset vividly illustrate these limitations, indicating the necessity for future research and model improvements to handle non-Lambertian surfaces.

VI. CONCLUSION

This paper proposed a neural network-based method for refining noisy disparity maps produced by any traditional or deep learning-based stereo algorithm. Through careful experimentation, we have identified key factors that contribute to the effectiveness of our method, including the use of advanced feature extractors and data augmentation techniques. Our method performs well on a variety of depth representations and scenarios, achieving strong generalization performance on real-world datasets despite being trained exclusively on synthetic data. Moreover, it can improve the quality of depth maps produced by traditional multi-view stereo methods and estimate the underlying aleatoric uncertainty. Therefore, our framework represents a promising solution for improving the accuracy and reliability of depth maps in a wide range of applications, including unbalanced stereo setups typically found in mobile devices.

REFERENCES

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1-3, pp. 7–42, 2002.
[2] M. Poggi, F. Tosi, K. Batsos, P. Mordohai, and S. Mattoccia, "On the synergies between machine learning and binocular stereo for depth estimation from images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5314–5334, Sep. 2022.
[3] N. Mayer et al., "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4040–4048.

[4] A. Kendall et al., "End-to-end learning of geometry and context for deep stereo regression," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 66–75.

[5] L. Lipson, Z. Teed, and J. Deng, "RAFT-stereo: Multilevel recurrent field transforms for stereo matching," in *Proc. Int. Conf. 3D Vis.*, 2021, pp. 218–227.

[6] J. Li et al., "Practical stereo matching via cascaded recurrent network with adaptive correlation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16 263–16 272.

[7] F. Tosi, Y. Liao, C. Schmitt, and A. Geiger, "SMD-nets: Stereo mixture density networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8938–8948.

[8] F. Aleotti et al., "Neural disparity refinement for arbitrary resolution stereo," in *Proc. Int. Conf. 3D Vis.*, 2021, pp. 207–217.

[9] M. Poggi et al., "On the confidence of stereo matching in a deep-learning era: A quantitative evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5293–5313, Sep. 2022.

[10] S. Gidaris and N. Komodakis, "Detect, replace, refine: Deep structured prediction for pixel wise labeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5248–5257.

[11] K. Batsos and P. Mordohai, "RecResNet: A recurrent residual CNN architecture for disparity map enhancement," in *Proc. Int. Conf. 3D Vis.*, 2018, pp. 238–247.

[12] M. Ferrera, A. Boulch, and J. Moras, "Fast stereo disparity maps refinement by fusion of data-based and model-based estimations," in *Proc. Int. Conf. 3D Vis.*, 2019, pp. 9–17.

[13] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 501–518.

[14] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[15] J. Zbontar et al., "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2287–2318, 2016.

[16] S. D. Cochran and G. Medioni, "3-D surface description from binocular stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 10, pp. 981–994, Oct. 1992.

[17] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 4, pp. 401–406, Apr. 1998.

[18] K. Mühlmann, D. Maier, J. Hesser, and R. Männer, "Calculating dense disparity maps from color stereo images, an efficient implementation," *Int. J. Comput. Vis.*, vol. 47, no. 1, pp. 79–88, 2002.

[19] S. Perreault and P. Hébert, "Median filtering in constant time," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2389–2394, Sep. 2007.

[20] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 49–56.

[21] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter (WMF)," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2830–2837.

[22] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[23] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, 1998, pp. 839–846.

[24] P. Favaro, "Recovering thin structures via nonlocal-means regularization with application to depth from defocus," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1133–1140.

[25] H. Kwon, Y.-W. Tai, and S. Lin, "Data-driven depth map refinement via multi-scale sparse representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 159–167.

[26] S. R. Fanello et al., "Filter forests for learning data-dependent convolutional kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1709–1716.

[27] M. Poggi, F. Tosi, and S. Mattoccia, "Quantitative evaluation of confidence measures in a machine learning world," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5228–5237.

[28] F. Tosi, M. Poggi, and S. Mattoccia, "Leveraging confident points for accurate depth refinement on embedded systems," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 158–167.

[29] M. Rossi, M. E. Gheche, A. Kuhn, and P. Frossard, "Joint graph-based depth refinement and normal estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12 154–12 163.

[30] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 878–886.

[31] Z. Liang et al., "Learning for disparity estimation through feature constancy," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2811–2820.

[32] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia, and L. D. Stefano, "Real-time self-adaptive deep stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 195–204.

[33] T. Saikia, Y. Marrakchi, A. Zela, F. Hutter, and T. Brox, "AutoDispNet: Improving disparity estimation with AutoML," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1812–1823.

[34] X. Song, X. Zhao, H. Hu, and L. Fang, "EdgeStereo: A context integrated residual pyramid network for stereo matching," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 20–35.

[35] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia, "SegStereo: Exploiting semantic information for disparity estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 636–651.

[36] Z. Yin, T. Darrell, and F. Yu, "Hierarchical discrete distribution decomposition for match density estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6044–6053.

[37] V. Tankovich, C. Hane, Y. Zhang, A. Kowdle, S. Fanello, and S. Bouaziz, "HITNet: Hierarchical iterative tile refinement network for real-time stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14 362–14 372.

[38] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5410–5418.

[39] S. Khamis, S. Fanello, C. Rhemann, A. Kowdle, J. Valentin, and S. Izadi, "StereoNet: Guided hierarchical refinement for real-time edge-aware depth prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 573–590.

[40] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, "GA-Net: Guided aggregation net for end-to-end stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 185–194.

[41] X. Cheng, P. Wang, and R. Yang, "Learning depth with convolutional spatial propagation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2361–2379, Oct. 2020.

[42] X. Cheng et al., "Hierarchical neural architecture search for deep stereo matching," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 1858.

[43] S. Duggal, S. Wang, W.-C. Ma, R. Hu, and R. Urtasun, "DeepPruner: Learning efficient stereo matching via differentiable PatchMatch," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4384–4393.

[44] G. Yang, J. Manela, M. Happold, and D. Ramanan, "Hierarchical deep stereo matching on high-resolution images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5515–5524.

[45] Y. Wang et al., "Anytime stereo image depth estimation on mobile devices," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 5893–5900.

[46] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, "Group-wise correlation stereo network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3273–3282.

[47] Z. Shen, Y. Dai, and Z. Rao, "CFNet: Cascade and fused cost volume for robust stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13 906–13 915.

[48] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 402–419.

[49] Z. Li et al., "Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6197–6206.

[50] W. Guo et al., "Context-enhanced stereo transformer," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2022, pp. 263–279.

[51] C. Godard, O. MacAodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6602–6611.

[52] Y. Zhong, H. Li, and Y. Dai, "Open-world stereo video matching with deep RNN," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 101–116.

[53] A. Tonioni, M. Poggi, S. Mattoccia, and L. D. Stefano, "Unsupervised domain adaptation for depth prediction from images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2396–2409, Oct. 2020.

[54] F. Aleotti, F. Tosi, L. Zhang, M. Poggi, and S. Mattoccia, "Reversing the cycle: Self-supervised deep stereo through enhanced monocular distillation," in *Proc. 16th Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 614–632.

[55] S. Joung, S. Kim, K. Park, and K. Sohn, "Unsupervised stereo matching using confidential correspondence consistency," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 2190–2203, May 2020.

[56] H.-Y. Lai, Y.-H. Tsai, and W.-C. Chiu, "Bridging stereo matching and optical flow via spatiotemporal correspondence," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1890–1899.

[57] M. Poggi, A. Tonioni, F. Tosi, S. Mattoccia, and L. D. Stefano, "Continual adaptation for deep stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4713–4729, Sep. 2022.

[58] C. Zhang, K. Tian, B. Fan, G. Meng, Z. Zhang, and C. Pan, "Continual stereo matching of continuous driving scenes with growing architecture," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18 901–18 910.

[59] H. Wang, X. Wang, J. Song, J. Lei, and M. Song, "Faster self-adaptive deep stereo," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 175–191.

[60] C. Cai, M. Poggi, S. Mattoccia, and P. Mordohai, "Matching-space stereo networks for cross-domain generalization," in *Proc. Int. Conf. 3D Vis.*, 2020, pp. 364–373.

[61] F. Zhang, X. Qi, R. Yang, V. Prisacariu, B. Wah, and P. Torr, "Domain-invariant stereo matching networks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 420–439.

[62] M. Poggi, D. Pallotti, F. Tosi, and S. Mattoccia, "Guided stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 979–988.

[63] L. Bartolomei, M. Poggi, F. Tosi, A. Conti, and S. Mattoccia, "Active stereo without pattern projector," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 18424–18436.

[64] J. Zhang et al., "Revisiting domain generalized stereo matching networks from a feature consistency perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13 001–13 011.

[65] B. Liu, H. Yu, and G. Qi, "GraftNet: Towards domain generalized stereo matching with a broad-spectrum and task-oriented feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13 012–13 021.

[66] W. Chuah, R. Tennakoon, R. Hoseinnezhad, A. Bab-Hadiashar, and D. Suter, "ITSA: An information-theoretic approach to automatic shortcut avoidance and domain generalization in stereo matching networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13 022–13 032.

[67] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1623–1637, Mar. 2022.

[68] F. Tosi, A. Tonioni, D. De Gregorio, and M. Poggi, "NeRF-supervised deep stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 855–866.

[69] X. Hu and P. Mordohai, "A quantitative evaluation of confidence measures for stereo vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2121–2133, Nov. 2012.

[70] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4455–4465.

[71] J. J. Park, P. Florence, J. Straub, R. A. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 165–174.

[72] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5932–5941.

[73] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2304–2314.

[74] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 626.

[75] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3501–3512.

[76] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 523–540.

[77] L. Zhu et al., "RGB-D local implicit function for depth completion of transparent objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4649–4658.

[78] M. Oechsle, L. Mescheder, M. Niemeyer, T. Strauss, and A. Geiger, "Texture fields: Learning texture representations in function space," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4530–4539.

[79] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.

[80] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "GRAF: Generative radiance fields for 3D-aware image synthesis," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 1692.

[81] A. Kirillov, Y. Wu, K. He, and R. B. Girshick, "PointRend: Image segmentation as rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9796–9805.

[82] Y. Chen, S. Liu, and X. Wang, "Learning continuous image representation with local implicit image function," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8628–8638.

[83] W. Jiang, E. Trulls, J. Hosang, A. Tagliasacchi, and K. M. Yi, "COTR: Correspondence transformer for matching across images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6207–6217.

[84] C. Chen, X. Chen, and H. Cheng, "On the over-smoothing problem of CNN based disparity estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8997–9005.

[85] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[86] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[87] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proc. 3rd Eur. Conf. Comput. Vis.*, Secaucus, NJ, USA, 1994, pp. 151–158.

[88] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5695–5703.

[89] Y. Cabon, N. Murray, and M. Humenberger, "Virtual KITTI 2," 2020, *arXiv:2001.10773*.

[90] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3061–3070.

[91] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[92] D. Scharstein et al., "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. German Conf. Pattern Recognit.*, Springer, 2014, pp. 31–42.

[93] T. Schöps et al., "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2538–2547.

[94] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*.

[95] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11 976–11 986.

[96] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10 428–10 436.

[97] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1492–1500.

[98] Y. Lee, J. Kim, J. Willette, and S. J. Hwang, "MPViT: Multi-path vision transformer for dense prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7287–7296.

[99] W. Xu, Y. Xu, T. Chang, and Z. Tu, "Co-scale conv-attentional image transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9981–9990.

[100] W. Yu et al., "Metaformer is actually what you need for vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10 819–10 829.

[101] F. Aleotti, M. Poggi, F. Tosi, and S. Mattoccia, "Learning end-to-end scene flow by distilling single tasks knowledge," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 10435–10442.

[102] H. Xu and J. Zhang, "AANet: Adaptive aggregation network for efficient stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1959–1968.

[103] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch stereo-stereo matching with slanted support windows," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 1–11.

[104] Z. Jie et al., "Left-right comparative recurrent model for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3838–3846.
[105] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.
[106] P. Schröppel, J. Bechtold, A. Amiranashvili, and T. Brox, "A benchmark and a baseline for robust multi-view depth estimation," in *Proc. Int. Conf. 3D Vis.*, 2022, pp. 637–645.
[107] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2017, pp. 2432–2443.
[108] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs, "Large scale multi-view stereopsis evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 406–413.
[109] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 78.
[110] P. Z. Ramirez, F. Tosi, M. Poggi, S. Salti, S. Mattoccia, and L. Di Stefano, "Open challenges in deep stereo: The booster dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 21136–21146.

Fabio Tosi received the PhD degree in computer science and engineering from the University of Bologna, in 2021. Currently, he is assistant professor with the Department of Computer Science and Engineering, University of Bologna. His research interests include deep learning and depth sensing related topics.

Filippo Aleotti received the master's degree in computer science and the PhD degree in structural and environmental health monitoring and management (SEHM2) from the University of Bologna, in 2018 and 2022, respectively. He has joined the Research & Development Team, Niantic, where he works on machine learning for augmented reality. His research interests include depth perception and scene understanding.

Pierluigi Zama Ramirez received the PhD degree in computer science and engineering, in 2021. He has been a research intern with Google for 6 months and is currently assistant professor with the University of Bologna. He co-authored 15 publications in international conferences and journals on a variety of computer vision research topics such as semantic segmentation, depth estimation, optical flow, domain adaptation, virtual reality, and 3D reconstruction.

Matteo Poggi (Member, IEEE) received the master's degree in computer science and the PhD degree in computer science and engineering from the University of Bologna, in 2014 and 2018, respectively. Currently, he is assistant professor with the Department of Computer Science and Engineering, University of Bologna. He co-authored 85 papers, mostly about deep learning for depth estimation and related tasks.

Samuele Salti is currently associate professor with the Department of Computer Science and Engineering (DISI), University of Bologna, Italy. His main research interest include computer vision, in particular 3D computer vision, and machine/deep learning applied to computer vision problems. He has co-authored more than 50 publications and eight international patents. In 2020, he co-founded the start-up eyecan.ai.

Stefano Mattoccia (Senior Member, IEEE) is currently an associate professor with the Department of Computer Science and Engineering, University of Bologna. His research activity concerns computer vision, mainly focusing on depth perception, and related tasks. In these fields, he co-authored more than 130 scientific publications.

Luigi Di Stefano (Member, IEEE) received the PhD degree in electronic engineering and computer science from the University of Bologna, in 1994. He is a full professor with the University of Bologna's Department of Computer Science and Engineering, leading the Computer Vision Laboratory (CVLab). His research interests include image processing, computer vision, and machine/deep learning. Author of more than 150 papers and several patents. He is a scientific consultant for major companies in the fields of computer vision and machine learning. He is a member of the IEEE Computer Society and the IAPR-IC.

Open Access funding provided by 'Alma Mater Studiorum - Università di Bologna' within the CRUI CARE Agreement