This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3070917, IEEE Transactions on Pattern Analysis and Machine Intelligence

On the Synergies between Machine Learning and Binocular Stereo for Depth Estimation from Images: a Survey

Matteo Poggi, *Member, IEEE,* Fabio Tosi, *Student Member, IEEE,*
Konstantinos Batsos, *Student Member, IEEE,*
Philippos Mordohai, *Member, IEEE,* and Stefano Mattoccia, *Member, IEEE*

Abstract—Stereo matching is one of the longest-standing problems in computer vision with close to 40 years of studies and research. Throughout the years the paradigm has shifted from local, pixel-level decision to various forms of discrete and continuous optimization to data-driven, learning-based methods. Recently, the rise of machine learning and the rapid proliferation of deep learning enhanced stereo matching with new exciting trends and applications unthinkable until a few years ago. Interestingly, the relationship between these two worlds is two-way. While machine, and especially deep, learning advanced the state-of-the-art in stereo matching, stereo itself enabled new ground-breaking methodologies such as self-supervised monocular depth estimation based on deep networks. In this paper, we review recent research in the field of learning-based depth estimation from single and binocular images highlighting the synergies, the successes achieved so far and the open challenges the community is going to face in the immediate future.

Index Terms—Stereo matching, machine learning, deep learning, monocular depth estimation

1 Introduction

Since the early stages of computer vision, estimating depth from images has been one of the iconic challenges for researchers. Obtaining dense and accurate depth maps is crucial for effectively addressing higher-level tasks such as 3D reconstruction, mapping and localization, autonomous driving, and many more. The focus of this paper is on stereo matching, which is classified as a passive sensing technique, and related topics. Competing technologies for depth estimation rely on active sensing which comes in several forms, including structured light projection, Time-Of-Flight (ToF) measurement, Laser Imaging Detection and Ranging (LIDAR) among others. Common to these devices is the perturbation of the environment required to sense depth. Although very accurate and precise, these sensors suffer from non-negligible weaknesses limiting their practical deployment for real applications. For instance, LIDAR sensors, which rely on one or more laser emitters scanning the environment through mechanical rotation, may suffer from misalignment, missing laser returns due to absorbing or reflective surfaces and multi-pathing. Moreover, they typically provide only sparse measurements of the observed scene, with density (and pricing) increasing with the number of laser emitters. For structured-light devices, such as the Microsoft Kinect, the pattern projection technology constrains the working range to a few meters and prevents usage under direct sunlight.

Inferring depth from images acquired by a regular camera has the potential to overcome all the limitations above. Among the different techniques for this purpose, stereo matching [1] takes as input two rectified images and attempts to compute the disparity of every pixel by matching corresponding pixels along conjugate epipolar lines, thus enabling depth estimation via triangulation. Years of research proved the effectiveness of stereo, making it a viable alternative to expensive active sensors often deployed in practical applications. The success and proliferation of machine learning and deep learning techniques in computer vision [2] led to notable improvements to stereo matching, even though it was one of the areas of computer vision in which learning was adopted relatively late. At the same time, the most recent advances in depth estimation from images have demonstrated that deep learning itself could benefit from stereo to achieve goals unimaginable just a few years ago, as in the case of self-supervised single-image depth estimation enabled through view synthesis [3] or other stereo-based strategies. Thus, the synergy between these two worlds led to outstanding results, shown in Fig. 1.

In this paper, we present a comprehensive review of the last years of progress in the field of depth estimation via binocular stereo matching and related topics. Starting from early attempts to leverage machine learning to replace single steps of the traditional stereo pipeline [1], we will guide the reader through five years of research, highlighting the successes achieved so far and pointing out the open challenges the community is going to face in the immediate

- *M. Poggi, F. Tosi and S. Mattoccia are with the Department of Computer Science and Engineering, University of Bologna, Italy, IT. {m.poggi,fabio.tosi5,stefano.mattoccia} @unibo.it*
- *K. Batsos is with Argo AI. kbatsos@stevens.edu*
- *P. Mordohai is with the Department of Computer Science, Stevens Institute of Technology, New Jersey, USA. philippos.mordohai@stevens.edu*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3070917, IEEE Transactions on Pattern Analysis and Machine Intelligence

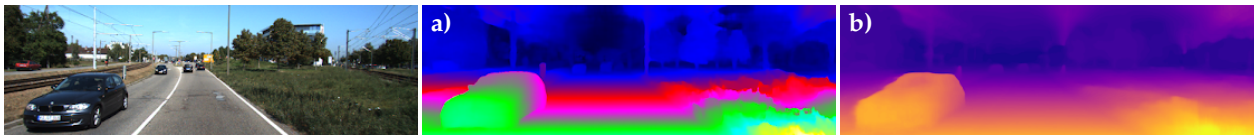

Fig. 1. **Years of progress in the field of stereo vision and machine learning** enable the estimation of depth maps of unprecedented quality from a) stereo or b) monocular images.

future. This paper extends the topics covered by *Learning-based depth estimation from stereo and monocular images: successes, limitations and future challenges* tutorials offered at 3DV 2018[1] and CVPR 2019[2]. We argue that it is timely since the previous surveys on stereo [1], [4] are outdated.

The rest of the manuscript is organized as follows: Section 2 introduces the most popular datasets and benchmarks in stereo matching, Section 3 discusses early attempts to replace individual steps of conventional stereo pipelines [1] with learning-based techniques, followed by Section 4 that reviews and classifies end-to-end models for stereo matching. Then, we consider two aspects concerning respectively the conventional pipelines and the end-to-end models, that are confidence estimation, covered in Section 5, and the domain-shift problem, introduced in Section 6 together with techniques aimed at mitigating it. Then, Section 7 reviews single-image depth estimation frameworks supervised by means of stereo images and finally, Section 8 collects take-home messages from our survey.

2 Datasets

In most computer vision problems, the availability of large and diverse datasets is of paramount importance for successfully developing new algorithms and for being able to measure their effectiveness. For years, researchers in stereo matching evaluated their proposals on a few dozen stereo pairs with ground truth depth maps acquired in controlled, indoor environments [1], [5], [6]. Although these datasets allowed notable progress in the design of stereo algorithms, they did not adequately highlight many of the challenges arising in real applications. Moreover, modern machine learning algorithms are data-hungry and require much more than a few dozen stereo pairs.

In 2012, the first large-scale dataset with images of outdoor, real environments was released [7] and an indoor dataset with much higher resolution [8] appeared soon after. Later, with the advent of deep learning [2] these datasets were followed by large, synthetic image sets which are ideal for training deep networks thanks to the negligible cost required to generate a multitude of training samples. In all cases, the datasets provide depth annotations obtained through different methodologies discussed later. The rest of this section will introduce in detail each of these datasets, summarized in Fig. 2 where we show one reference image and the associated ground truth disparity map for each of them, respectively for a) KITTI 2015, b) Middlebury 2014, c) ETH3D and d) Freiburg SceneFlow. The first three were the foundation of the stereo aspect of the Robust Vision Challenge (ROB)[3] in 2018.

1. sites.google.com/view/3dv-2018-depth-from-image
2. sites.google.com/view/cvpr-2019-depth-from-image
3. robustvision.net

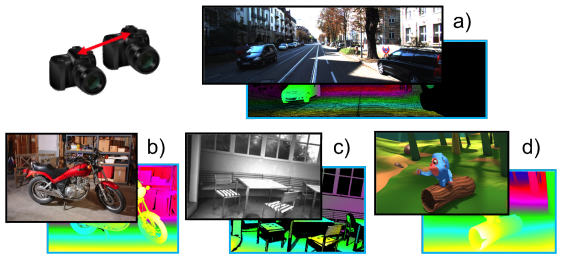

Fig. 2. **Overview of the most popular stereo datasets in literature**, with examples of reference images and associated ground truth disparity. a) KITTI 2015 [9], b) Middlebury 2014 [8], c) ETH3D [10], d) Freiburg SceneFlow [11].

2.1 KITTI

Acquired by Geiger *et al.* [12], the KITTI Vision Benchmark Suite represents the first, large-scale collection of images from a driving environment. The KITTI benchmarks have been seminal to the development of several algorithms and methods supporting autonomous driving. The data have been acquired from a car equipped with two stereo camera pairs, one grayscale and one color, a Velodyne LIDAR, GPS and inertial sensors. It consists of about 42k stereo pairs and LIDAR point clouds taken from 61 different scenes. From this extensive collection of images, appropriate benchmarks are available for key computer vision tasks such as stereo, optical flow, visual odometry, object detection and more. Two main datasets are available for stereo matching: KITTI 2012 and KITTI 2015.

KITTI 2012 [7]. This is the first dataset for stereo matching comprising outdoor images of static scenes and providing an online benchmark[4] for evaluation. It consists of 389 grayscale stereo pairs (recently made available in color format as well), split into 194 training pairs with available ground truth and 195 test pairs with withheld ground truth. Ground truth depth was obtained from LIDAR measurements as follows. A set of consecutive frames (5 before and 5 after) were registered using ICP, accumulated point clouds were re-projected onto the image, and finally, all ambiguous image regions such as windows and fences were manually removed. Using calibration parameters, the 3D points were projected on the images to obtain depth measurements, which were converted into disparities. This strategy yields semi-dense ground truth maps, covering about one third of the pixels in each input image. The error metrics on the benchmark are the percentage of pixels with a disparity error greater than 3 and the average disparity error, measured either on all pixels or non-occluded pixels only. In both cases, the lower the better. Metrics computed in reflective regions are also available.

4. cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo

KITTI 2015 [9]. A few years later, an improved dataset and benchmark for scene flow estimation [9] was proposed. In this case, the dataset consists of 400 color stereo pairs, evenly split into training and test sets. In contrast to the previous dataset, the stereo pairs are from dynamic scenes with objects (mostly cars) moving independently. The same procedure used for KITTI 2012 is followed here to obtain ground truth labels, except for moving objects whose 3D points cannot be properly accumulated over time. Hence, to obtain depth annotations for cars, 3D cad models are fitted into accumulated point clouds and re-projected onto the image. As the primary evaluation metric, the percentage of pixels with an absolute disparity error greater than 3 and a relative error larger than 5% (D1) is reported on the online benchmark[5], the lower the better. The D1 metric is listed for foreground (*i.e.* belonging to moving objects), background or all pixels. Moreover, masks to distinguish between non-occluded and all pixels are available.

2.2 Middlebury

The Middlebury Stereo Vision Page provided the first benchmark that allowed authors to submit the results of their algorithms. Over the years, the Middlebury stereo datasets have provided indoor images with dense ground truth labels, obtained by manual annotation at first [1] and by structured light sensors later [5], [6], [8]. Three main versions have been proposed between 2002 [1] and 2014 [8], with varying resolution and image content. We will focus on this latter version, namely *Middlebury 2014*, since it provides an online benchmark for evaluation and still represents one of the most challenging datasets for stereo matching.

Middlebury 2014 [8]. It consists of 33 scenes, divided into *training*, *additional* and *test* splits made of respectively 13, 10 and 10 stereo pairs. Some of the data are used multiple times under different exposure and illumination conditions. A unique feature of this dataset is the very high image resolution, which reaches 6 megapixels compared to 0.3 megapixels of the KITTI images, and a disparity range between 200 and 800 pixels, representing one of the hardest challenges of this dataset. Images and ground truth disparity maps are provided at full (F), half (H) and quarter (Q) resolution. An active stereo pipeline, described in detail in [8], was deployed to obtain dense and accurate ground truth depth. The limited number of training samples and the variety of content in the images make this dataset particularly challenging for deep learning methods, in particular for end-to-end models as we will set in the next sections. The online benchmark[6], reports the percentage of pixels having disparity errors larger than 0.5, 1, 2 and 4, as well as average and root mean square errors (RMSE) and other metrics on either all or non-occluded pixels.

2.3 ETH3D

ETH3D [10] is a recent, real-world, multi-view dataset for 3D reconstruction acquired in both indoor and outdoor environments at ETH Zurich. It consists of 25 high-resolution color multi-view stereo scenes divided into 13 for training and 12 for testing, 10 low-resolution grayscale many-view videos evenly divided for training and testing and finally 47 low-resolution grayscale stereo pairs, respectively split into 27 and 20 for training and testing. To obtain ground truth disparities, the authors recorded the scene geometry with a Faro Focus X 330 laser scanner, taking one or more 360° scans with up to 28 million points each. Together with depth, the color of each 3D point captured by the laser scanner's integrated RGB camera was acquired, taking about 9 minutes to collect a single scan. The online benchmark[7], reports similar metrics to those of the Middlebury 2014 dataset.

2.4 Freiburg SceneFlow

The Freiburg SceneFlow dataset [11], [13] was a groundbreaking step forward in the field. As evidence, we underline that most of the proposed end-to-end networks for stereo matching are trained from scratch on this large dataset, before being fine-tuned on real data. The dataset consists of 3D scenes, from which images and dense ground truth for stereo, optical flow, and scene flow are rendered. To this end, the authors modified the internal rendering engine of the freely available Blender suite in order to produce fully dense and accurate ground truth for the two views of a virtual stereo camera with a resolution of 540×960 pixels. The dataset is organized into three subsets, named *FlyingThings3D*, *Monkaa* and *Driving*, totalling about 39000 stereo pairs overall. We briefly summarize the three datasets, referring the reader to [13] for more details.

FlyingThings3D. This set of images has been obtained fully automatically: the authors created a structured background from random geometric shapes, and overlayed on it dynamic foreground objects sampled from ShapeNet [14] and following linear trajectories in 3D space, as the camera itself does. It totals 22872 stereo pairs, while 4370 more are set aside as the validation set of the full SceneFlow dataset.

Monkaa. In contrast to FlyingThings3D, stereo pairs contained in this split are generated from an animated movie in a deterministic way. 3D artists modeled original scenes and elements, then the authors produced custom environments and rendered long scenes to sample sufficient data. This subset contains 8591 stereo pairs.

Driving. Similarly to Monkaa, this split has been generated in a deterministic way as well. The aim of this portion of the Freiburg dataset is to provide data relevant to driving environments, as opposed to general scenes. This set contains 4392 samples.

2.5 Other datasets

We mention a few more datasets which have been used less widely with respect to the previous ones.

MPI-Sintel [15]. Originally proposed for dense optical flow benchmarking, the Sintel dataset is a collection of synthetic images extracted from short, animated movies. Together with dense flow labels, it recently has made available stereo pairs, disparity and occlusion ground truth annotations for 23 scenes with a total of 1109 stereo pairs. However, this dataset has rarely been used in the evaluation

5. cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo
6. vision.middlebury.edu/stereo/eval3
7. eth3d.net/low_res_two_view

of stereo approaches [16], [17]; it is much more popular in optical flow.

CARLA [18]. This framework implements a simulator allowing for data generation in the context of autonomous driving. It provides open-source code and digital assets (*e.g.* urban layouts, buildings, vehicles), allowing for synthesis of virtual environments in varying weather conditions and with full control of static and dynamic actors in the scene. Agents with custom sensors, such as a stereo camera, can navigate in the simulated world and acquire a potentially unlimited amount of images with dense annotations. Although rarely used to train stereo networks [19], the aforementioned modelling power makes it a promising tool for future research.

More driving datasets. We conclude listing newly-proposed imagery acquired in driving environments. The Oxford Robotcar dataset [20] has been acquired after more than 100 km navigation with a trinocular camera, thus collecting stereo pairs with both a narrow and a wide baseline. Ground truth depth is generated from raw LIDAR measurements. Apolloscape [21] provides 5165 stereo pairs at 3 megapixels resolution, divided into 4156 pairs for training and 1009 for testing, with dense ground truth obtained by point cloud accumulation and fitting 3D CAD models, similarly to KITTI 2015. The recent DrivingStereo dataset [22] provides over 180k stereo pairs at 1.4 megapixels resolution, with semi-dense ground truth disparities obtained by interpolating LIDAR measurements and refining them with a deep stereo network. Potentially relevant are very large datasets released to support research on autonomous driving. Specifically, the Waymo Open Dataset [23], Argoverse by Argo AI [24] and the Lyft Level 5 dataset [25] are of unprecedented scale and one could imagine rectified stereo pairs with ground truth being extracted from them. We anticipate that Apolloscape and DrivingStereo, which provide binocular stereo imagery directly, will play a significant role for future developments in stereo matching.

3 Learning within the Stereo Pipeline

Despite the proliferation and success of deep learning [2] in most high-level tasks in computer vision, low-level vision problems were only partially affected at the very beginning.

The initial research efforts applying machine learning in stereo vision aimed at improving individual steps of the established pipeline [1], for instance by learning a matching cost function to replace hand-crafted ones based on SAD or the census transform [28] or by learning how to improve the subsequent optimization and refinement stages after the conventional winner-takes-all (WTA) strategy. This first step ignited the rapid evolution of stereo algorithms of the last five years, progressively developing more robust methods as shown in Fig. 3.

3.1 Matching cost

Since stereo matching aims to detect correspondences between pixels, intuitively learning a robust matching function is a promising first step. Better matching costs also allow for improved volume optimization and thus lead to more accurate disparity maps. Critical for this kind of approaches is the possibility of extracting large amounts of training data from a few hundred images. Since the goal is learning correspondences between pixels, each pixel with available ground truth represents a training sample. This means that a relatively small dataset, such as KITTI 2015, provides more than 30 million samples, even though only 30% of the total pixels are labelled.

MC-CNN [27]. The most impactful work in this area is by Žbontar and LeCun who train a CNN to predict whether two image patches match or not. A Siamese network extracts features from the two images, which are passed to a fully connected network estimating a matching score for the center pixel of the left patch. By replacing fully connected layers with 1×1 convolutions, the architecture can be made fully convolutional to process the entire image at once. Two versions were developed: *MC-CNN-acrt* for which the features are concatenated, thus D forwards are required at test time (where D is the disparity range) and *MC-CNN-fst* which replaces concatenation with a dot product, allowing for a single forward pass through the network at the cost of a small drop in accuracy. In order to achieve state-of-the-art results, the cost volume obtained by MC-CNN is optimized and refined using a conventional SGM pipeline [29] including Cross Based Cross Aggregation (CBCA) [30].

Deep Embed [31]. In conventional pipelines, the choice of window size is crucial to the effectiveness of local aggregation. In particular, large windows allow for processing more information and are more robust to textureless regions, but produce blurred boundaries near depth discontinuities. Conversely, small windows are preferred near edges but are ineffective in ambiguous regions. Following this observation, Chen *et al.* design a network to learn a multi-scale feature embedding, processing 13×13 patches at full and half resolution, thus learning a cost function from both small and large windows. Final matching scores are obtained as the dot product between left and right multi-scale features, extracted by a Siamese feature extractor.

Content CNN [32]. The aforementioned approaches process patches separately, producing a score for each patch comparison that is independent of other comparisons. Luo *et al.* pose the problem as multi-class classification, where the classes are all possible disparities, instead of binary classification for each disparity. This leads to calibrated scores for each disparity and higher accuracy. The dot product, as in MC-CNN-fst, is used to combine left and right features.

Per-pixel pyramid-pooling [33]. Park and Lee enable the network to access wider context by adding a pyramid pooling layer that considers data over multiple scales without loss of resolution and detail. This leads to disparity maps with precise discontinuities and higher accuracy than MC-CNN-acrt, especially when avoiding SGM optimization.

SDC [34]. Schuster *et al.* propose a novel architecture for learning an universal descriptor for dense matching. By leveraging parallel dilated convolutions, with different dilation factors, SDC extracts features by processing a large receptive field with moderate increase of the computational cost. This solution is effective at improving performance of stereo, optical flow and scene flow algorithms when replacing traditional descriptors.

Consistency and distinctiveness [35]. Zhang and Wah argue that almost all existing problems in dense matching

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3070917, IEEE Transactions on Pattern Analysis and Machine Intelligence

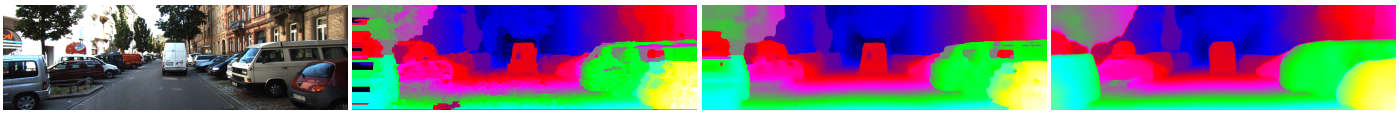

Fig. 3. **Evolution of stereo algorithms.** From left, reference image from KITTI 2015, disparity maps by SGM [26], MC-CNN-acrt [27] and DispNetC [11]. Learned matching costs outperform traditional pipelines, while end-to-end models perform even better in challenging regions (*e.g.* cars).

are caused by features that violate the principle of consistency, the principle of distinctiveness or both. Consistency requires that a given point should have similar descriptors when it is observed from different viewpoints. Distinctiveness states that a feature should be different from other pixels in its surrounding regions. The author seek guide features in a deep multi-objective optimization framework incorporating both principles.

CBMV [36]. Batsos *et al.* propose a method for learning the matching volume leveraging both data with ground truth and conventional wisdom. A random forest classifier determines the likelihood of whether a given disparity for a pixel is correct based on a combination of hand-crafted matching functions and long-range constraints. The resulting cost volume is optimized as in [27], leading to similar accuracy when testing in the training domain, but much better generalization across different domains.

Weakly-supervised deep metric [37]. Learned matching functions achieve high accuracy, but require substantial amounts of annotated data for training. Tulyakov *et al.* propose an effective strategy to leverage coarse information from the stereo setup, such as epipolar, uniqueness, smoothness and ordering constraints, to obtain weak supervision from stereo pairs with spare or no ground truth. Despite the weak supervision, the learned matching functions perform as well as those trained conventionally.

3.2 Optimization

After initial cost volume computation, optimization is crucial for gathering information from a larger context and overcoming the limitations of pixel-wise matching. SGM [26] is by far the most popular conventional technique for cost volume optimization; as a result, improving it via learning has received attention from the research community.

GCP [38], [39]. Based on the assumption that reliable pixels can be used to influence neighboring pixels within a global optimization framework, Spyropoulos *et al.* select highly reliable pixels, detected by a random forest classifier, as ground control points (GCPs). GCPs are, then, used to introduce soft constraints into the matching volume, which is optimized using MRF energy minimization [40].

LevStereo [41], [42]. Park and Yoon propose a generalized modulation strategy in order to improve the robustness and the accuracy of widely used stereo matching algorithms such as SGM. More specifically, cost curves of an initial cost volume showing evidence of low confidence values are flattened while highly confident pixels are left unchanged. This modulation scheme is effective because it enhances the importance of reliable matching costs inside the SGM aggregation step, allowing reliable pixels to guide disparity estimation for unreliable ones.

O1 [43], [44]. By analyzing in depth the SGM algorithm and observing that the Scanline Optimization (SO) strategy causes streaking artifacts in the final disparity map, Poggi and Mattoccia propose a more effective measure computed on features extracted from the disparity map only in constant time. Specifically, the standard SO scheme is replaced with a smarter strategy that properly weights the matching costs computed for each independent path using the corresponding confidence score. This leads to visible artifacts in the disparity map being considerably alleviated.

PBCP [45]. Seki and Pollefeys argue that not all pixels should be subject to the same smoothness penalties in SGM optimization. If penalties were decreased at the most confident pixels, scanline optimization would propagate information from reliable to unreliable pixels. This can be achieved by changing the SGM formulation, by adjusting the smoothness penalty parameters per pixel according to confidence scores, estimated by a CNN processing the initial WTA left and right disparity maps.

SGM-Net [46]. Seki and Pollefeys extend PBCP to distinguish between positive and negative disparity transitions along the scanlines, since they signify different occlusion relationships. They introduce a new loss function, that includes path and neighbor costs by taking into account the cost of the disparity path over a scanline compared to the ground truth and transitions between neighboring pixels, respectively.

SGM-Forest [47]. Following the rationale in [43], Schönberger *et al.* develop a random forest classifier for improving the selection among disparity values for a pixel proposed by multiple scanlines in SGM. The classifier considers disparities and optimized costs per scanline to produce per-pixel scores, used both to combine the disparity hypotheses from the different scanlines, as well as to obtain confidence.

3.3 Refinement

The last step of the pipeline aims to refine the estimated disparity map. Traditionally, image processing techniques like median or bilateral filtering are used for this task, after a left-right consistency check. Recently, neural networks have been proposed to replace traditional image filters.

GDN [48]. Shaked and Wolf develop a multi-stage architecture, named L-ResMatch, that addresses cost volume refinement in its last stage. L-ResMatch begins with a residual network that learns a matching cost function. Then, traditional aggregation steps like CBCA and SGM [27] are applied. Finally, a Global Disparity Network (GDN) locally refines the optimized cost volume to further improve the quality of the final disparity map, while predicting a confidence estimate for each pixel at the same time.

Detect Replace Refine (DRR) [49]. Gidaris and Komodakis present the DRR algorithm, which decomposes label improvement in a detection, a replacement and a refinement step. DRR is based on the hypothesis that hard

mistakes should be detected and replaced, because correcting them does not depend on the wrong input estimates, while soft mistakes can be corrected by additive refinement. The authors show that further improvements can be achieved if the network is applied iteratively.

Order-based Surface Decision (OSD) [50]. Ye *et al.* extend DRR [49] by distinguishing among different failure modes of the matching process. Different strategies for each case, and corresponding sub-networks that implement them, are introduced. The resulting system is able to improve the outputs of a diverse set of matching algorithms on the Middlebury 2014 benchmark.

RecResNet [51]. Batsos and Mordohai apply a dense label correction algorithm, implemented as a recurrent, residual network, to an input disparity map estimated by a black box stereo algorithm. The output disparity map is generated based on the noisy input disparity map and the left image by applying a combination of residuals computed at multiple scales, to correct heterogeneous types of errors. The same network is applied recurrently to its own output to make further improvements.

LRCR [52]. Taking advantage of disparity estimates from both the left and the right view, the *Left-Right Comparative Recurrent* (LRCR) model embeds the left-right consistency check into a unified pipeline in order to improve the final disparity estimate. A soft attention mechanism, jointly with recurrent learning, is in charge of selecting areas in the images for refinement, thus guiding the network to correct errors mainly on unreliable initial depth estimates.

VN [53]. Departing from the more common refinement processes based on residual corrections, Knöbelreiter and Pock propose a learning-based model built on a variational refinement network for the same purpose. In particular, the network takes the RGB image, the initial disparity map and a confidence measure as input, and performs collaborative denoising, considering that errors can be identified by considering the three inputs jointly.

3.4 Experimental comparison

In this section, we report a quantitative comparison between the approaches that apply learning to stages of the stereo pipeline discussed so far. To ensure a fair comparison, we retrieve results from popular online benchmarks for stereo matching [7], [8], [9], [10]. Since not all methods have been submitted to the online benchmarks, we focus on the subset providing such results. If available, we select results not labelled *ROB* since training is likely to be on the data provided by the benchmark itself.

KITTI 2015 [9]. Table 1 shows results of methods covered in this section on the KITTI 2015 leaderboard. At the bottom of the table, we also report the results of SGM, as a non data-driven baseline, highlighted in yellow. First, we can see how all methods outperform SGM by a large margin. MC-CNN achieves a major boost in accuracy, while PBCP and SGM-Net that built upon it further improve in accuracy. Unsurprisingly, disparity refinement approaches attain the best results among methods reviewed in this section since they are applied on the output of other algorithms.

Middlebury 2014 [8]. Table 2 summarizes results from the Middlebury benchmark. In this case, we can see how

Method	KITTI 2015 [9]			
	D1-bg% (↓)	D1-fg% (↓)	D1-all% (↓)	time (s) (↓)
LRCR [54]	2.55	5.42	3.03	49.2
RecResNet [51]	2.46	6.30	3.10	1.3
DRR [49]	2.58	6.04	3.16	0.4
L-ResMatch [48]	2.72	6.95	3.42	48
PBCP [45]	2.58	8.74	3.61	68
SGM-Net [46]	2.66	8.64	3.66	67
MC-CNN-acrt [27]	2.89	8.88	3.89	67
SGM-Forest [47]	3.11	10.74	4.38	6
Content-CNN [32]	3.73	8.58	4.54	1
VN [53]	4.29	7.65	4.85	0.5
CBMV [36]	4.17	9.53	5.06	250
OpenCV-SGBM [26]	8.92	20.59	10.86	1.1

TABLE 1
KITTI 2015 leaderboard [9], showing methods learning stages of the pipeline.

Method	Middlebury 2014 [8]			ETH3D [10]	
	Res.	bad 2.0% (↓)	avg. px (↓)	bad 1.0% (↓)	avg. px (↓)
SGM-Forest [47]	H	7.37	2.84	4.96	0.36
MC-CNN-acrt [27]	H	8.08	3.82	-	-
CBMV [36]	H	11.1	4.71	5.35	0.33
VN [53]	H	14.2	2.49	-	-
SGM [26]	H	18.4	5.32	10.08	0.50

TABLE 2
Middlebury 2014 [8] and ETH3D [10] leaderboards, showing methods learning stages of the pipeline.

SGM-forest performs much better compared to what is observed on KITTI. Methods leveraging on deeper models, such as refinement techniques, do not appear on this online benchmark. The same is true for end-to-end models that we are going to discuss in the remainder. This is due to the small number of training images available for fine-tuning these more complex networks.

ETH3D [10]. We report, for the sake of completeness, results on the ETH3D in Table 2 although this benchmark was published later than the ones above. SGM-forest confirms its good performance on ETH3D as well.

4 End-to-end deep learning

Although machine learning substantially improved each step of the traditional stereo matching pipeline [1], the introduction of end-to-end models drove the community towards a new paradigm. As can be seen in popular benchmarks, KITTI 2012 [7] and 2015 [9], in just a few years, end-to-end methods dominated dense disparity estimation, thanks to the availability of large amounts of labeled data. Indeed, while few hundred annotated images [7], [8], [9] are enough to train learning-based pipeline stages, they are not for end-to-end models. However, thousands of labeled pairs can be made available for free using graphics [11], [13], overcoming the costly and cumbersome process of labeling a large number of images with accurate depth measurements.

The rest of this section is organized as follows: Section 4.1 introduces a taxonomy of deep models for end-to-end disparity estimation, divided in 2D and 3D architectures. We review the most relevant models for both classes in Section 4.2 and Section 4.3, respectively. Finally, Section 4.4 summarizes the performance of these networks on the benchmarks.

4.1 Taxonomy

According to the popular online benchmarks [7], [8], [9], [10], there are hundreds of published, and unpublished, deep networks competing for the top ranks. We can broadly

categorize them into two distinct classes according to their design: *2D architectures* and *3D architectures*. The main difference between the two is the strategy deployed to encode features and geometry. Next, we discuss the differences between the two categories, while introducing models from the literature belonging to both, highlighting their distinctive features and the rationale behind their design. For additional details, such as training schedules or layer configuration, we refer readers to the corresponding papers.

4.2 2D architectures

This family of deep networks is closer to neural models designed to solve other dense regression tasks such as semantic segmentation [55], optical flow [56] or monocular depth estimation [57]. These architectures usually deploy an encoder-decoder design, inspired by the U-Net model [58] to keep memory requirements and runtime manageable as well as to increase the receptive field of the network to leverage image context. Thanks to the efficiency of 2D convolution operations on modern GPUs, some of these models achieve from a few to dozens of frames per second at the cost of a negligible loss of accuracy [59], [60]. Pivotal to the spread of these architectures is the work by Mayer *et al.* [11] that introduced a custom layer, namely the *correlation layer*, in charge of computing similarity scores between features extracted from the two images.

DispNet-C [11]. The work by Mayer *et al.* [11] is a milestone for the switch to end-to-end disparity regression. Following the U-Net design [58], the authors propose DispNet-S, an encoder-decoder architecture. The first portion of the architecture feeds the input images to several 2D convolutional layers that decimate the input resolution. Then, stacks of 2D deconvolution layers gradually restore the resolution up to half the original, matching the full resolution using bilinear upsampling. In order to preserve fine details that were lost during downsampling, features from the encoding module are concatenated to corresponding ones at the same resolution extracted by the decoder. Due to this design choice, the network is large enough to learn disparity inference but is not trained to reason about correspondences explicitly. This latter behavior is attained by introducing a *correlation layer*, previously proposed in [56], computing similarity between patches x_1 and x_2 on the two images:

$$c(x_1, x_2) = \sum_{o \in [-k,k] \times [-k,k]} \langle f_1(x_1 + o), f_2(x_2 + o) \rangle \qquad (1)$$

with the two patches of size $K = 2k+1$. Despite the general formulation, K is typically set to 0, thus computing single-pixel correlations. A siamese sub-network extracts features from both images at the beginning. Then, correlating each patch in the reference image and all candidates in a $(2D+1)$ search range, yields an equal number of correlation maps, stacked along the channels dimension and forwarded to subsequent 2D convolutional layers. According to [11], introducing the correlation layer significantly improves the quality of estimated disparity maps, in particular on real datasets such as KITTI 2012 and 2015.

CNN+CRF [61]. Although not strictly an end-to-end network, we include the work by Knöbelreiter *et al.* here. The authors design a joint CNN and Conditional Random Field (CRF) model to infer dense disparity maps. Joint training is made possible by formulating the CRF as a maximum margin Markov network. We consider this as an interesting intermediate step between models aimed at implementing single steps in the pipeline and a fully end-to-end CNN.

CRL [62]. Although DispNet-C fails to outperform conventional hand-engineered stereo pipelines such as [27], it paved the way for the deployment of more sophisticated deep models. The first 2D architecture to rise in the KITTI leaderboards is the one by Pang *et al.* . Starting from the observation that learning a residual signal [63] is usually easy for a neural network, the authors combine a DispNet-C model with a second subnetwork, referred to as DispResNet, which is in charge of computing residual corrections to the initial disparity map, d_1, estimated by the DispNet-C subnetwork. To do so, d_1 is used to warp the right input image towards the left. The left image, the warped right image and their difference are fed to the DispResNet module which determines the adjustment to d_1.

iResNet [64]. Liang *et al.* develop a deep network, inspired by the different steps in the conventional stereo pipeline. The first portion extracts features at multiple scales and feeds them to a correlation layer, producing an initial disparity estimate. Then, as in CRL, this estimate is used to warp right-image features to the left camera, before being passed to a refinement module made of a new correlation layer with smaller search range and additional 2D convolution and deconvolution operators. This module is stacked multiple times for iterative refinement, and the architecture is dubbed the Iterative Residual Network (iResNet). iResNet ranked first in the 2018 Robust Vision Challenge.

DispNet-CSS [16]. Following the successful findings about residual learning, Ilg *et al.* address occlusions, motion and depth boundary estimation. Specifically, a stack of networks iteratively produces improved predictions, which are followed by subsequent residual refinements. Either a DispNet-CSS or a FlowNet-CSS can be trained for stereo matching or optical flow estimation, respectively, based on this scheme. In both cases, only the first subnetwork deploys a correlation layer. Combining the two CSS architectures, the authors also estimate scene flow.

AutoDispNet-CSS [17]. While a general-purpose encoder-decoder can perform reasonably well if trained for a specific task, the literature (and this survey as well) supports the thesis that careful design choices and tuning are necessary to obtain a state-of-the-art model. These can be accomplished automatically using *auto machine learning* (autoML) to optimize for the best stereo matching architecture. By defining a set of candidate operations, *e.g.* 2D convolutions and upsampling layers, Saikia *et al.* use the gradient-based method DARTS [65] to explore the space of architectures and find the best configuration for DispNet. Despite deploying only a subset of the possible design choices among the candidate layers, the obtained AutoDispNet-CSS performs comparably to state-of-the-art.

MADNet [60]. Although most of the architectures focus on accuracy, efficiency is also crucial, in particular in real-world applications. Tonioni *et al.* apply a coarse-to-fine strategy within their *Modularly Adaptive Network* (MADNet). Starting from the coarser level of a feature pyramid ex-

tracted from each image, features are fed to a correlation layer, then an initial disparity map is predicted by a 2D convolutional decoder and upsampled to the previous level of the pyramid. This predicted disparity is used to warp the right features to the left view, feeding them to another correlation layer with a small search range to estimate a better disparity map. The procedure continues until the highest resolution, where a refinement network with 2D dilated convolutions produces the final disparity map.

HD^3 [59]. Along these lines, a coarse-to-fine strategy has been exploited by other authors. Yin *et al.* couple it with discrete distribution estimation, which is used to estimate the degree of uncertainty of the predicted values, an aspect typically neglected by the regression approaches discussed so far. Following a pyramidal approach similar to [60], a correlation layer is fed with left and warped right features and a decoder extracts features in the density embedding space in order to predict a final match density. The conversion between dense distributions and motion fields (either flow or disparities) and vice-versa is necessary to handle upsampling and to generate supervision for the next level.

SegStereo [66]. Multi-task learning [67] has gained popularity in various areas of computer vision, where joint learning of multiple tasks leads to overall improvement in all tasks. Following this rationale, Yang *et al.* propose SegStereo for joint disparity estimation and semantic segmentation. The network extracts a representation with shared features for both tasks. The features are used as input to a correlation layer and a segmentation subnetwork. The output correlation maps and semantic embeddings are concatenated and forwarded to an encoder-decoder producing the disparity map. Semantic masks are obtained for both the left and right views and forced to be consistent by warping them according to the predicted disparity. Tackled together, both tasks achieve improved results.

EdgeStereo [68], [69]. Depth discontinuities represent some of the most challenging regions in the image for stereo matching. Evidence for this can be observed by looking at qualitative examples of disparity maps from the KITTI online benchmarks [7], [9]. Motivated by this evidence, Song *et al.* jointly learn disparity estimation and edge segmentation within their EdgeStereo framework. An edge detection subnetwork extracts a set of features used to estimate an edge map. They are processed together with conventional correlation scores to obtain the final disparity map, improving accuracy near depth discontinuities.

DSNet [70]. Following the multi-task trend, Zhan *et al.* propose a network with an encoder which is shared between semantic segmentation and disparity estimation, and is trained in a multi-stage manner. At first, it is optimized for semantic segmentation; then, the weights are fixed and used to learn disparity estimation by means of a matching module combining feature correlation and concatenation by means of attention mechanisms. Finally, a third stages optimizes the model for both tasks jointly.

SENSE [71]. Another step in the direction of multi-task learning is performed by Jiang *et al.* , proposing a single, compact architecture estimating disparity, optical flow, disparity change and semantic segmentation at once. Starting from a single, shared encoder, different decoders are introduced for each task.

Unsupervised Stereo [72]. Although they achieve compelling results, deep networks are heavily dependant on the amount and variety of the labeled training data. To simplify the training process of deep stereo networks, Zhou *et al.* propose an end-to-end framework capable of learning disparity estimation in an unsupervised manner. A 2D network with an image-guided aggregation network is designed to estimate disparity maps for the left and right images. Then, left-right consistent matches are used to train the network iteratively on its own predictions. Moreover, the proposed architecture has accuracy comparable to DispNet-C, when trained with supervision.

4.3 3D architectures

Whereas 2D networks are much closer to traditional neural models, 3D architectures were developed specifically for stereo matching. Although the traditional encoder-decoder design is embodied by these frameworks as well, they differ from the previous category by explicitly encoding geometry during the processing of the features. Conversely to 2D models, 3D networks explicitly encode matching properties between pixels in the form of feature vectors utilizing different operators, *e.g.* concatenation and feature difference. By performing this operation on the entire search range D, 3D architectures produce an output volume of increased dimensionality: $D \times H \times W$, times the amount of features F, resulting in a 4D data structure. Subsequently, the 4D tensor is processed by 3D convolutions, resulting in explicit processing of a matching volume-like representation. This strategy comes at the cost of a much higher memory requirements and runtime.

GC-Net [73]. This framework represents the first attempt to deploy explicit knowledge about geometry to design a 3D neural network for stereo matching. It also was the first end-to-end model to outperform hand-crafted pipelines on the KITTI benchmarks. High-level features are extracted from both images using two encoders with shared weights. In this phase, the original resolution is halved to reduce memory requirements and then a cost volume is built by concatenating per-pixel features F across the two images on the entire disparity search range, producing a $\frac{D}{2} \times \frac{W}{2} \times \frac{H}{2} \times 2F$ volume. Then, a 3D encoder-decoder module processes the volume to obtain a final $D \times H \times W \times 1$ volume, from which the disparity map is obtained using the *soft-argmin* operator

$$\text{soft-argmin} = \sum_{d=0}^{D} d \cdot \sigma(-c_d) \tag{2}$$

with σ the softmax operator applied to each final feature c_d along the D dimension.

ECA [74]. Following this successful, new design paradigm, several authors focused on further boosting the accuracy of 3D networks. Yu *et al.* propose to improve the 3D optimization phase by introducing *Explicit Cost Aggregation* (ECA) modules along the three different dimensions. This goal was achieved by adding a set of 3D convolutions having rectangular filters, with kernel size equal to 1 on all but one dimensions, keeping a low computational cost compared to traditional 3D convolutions. A further guided aggregation strategy is proposed, directly learning from the

image a set of guides to be applied to the final cost volume before the softargmin selection.

PSMNet [75]. Although an encoder-decoder structure allows for taking into account large context information during the learning process, in 3D networks this stage occurs only after the cost volume computation, which depends on very local features. Advances in deep learning introduced new layers capable of greatly enlarging the receptive field of a neural network with a negligible computational cost, as in the case of *Spatial Pyramidal Pooling layers* (SPP) [76] adopted by Chang and Chen in their *Pyramidal Stereo Matching network* (PSMNet). Integrating SPP layers in the GC-Net feature extractor, together with deploying a stack of multiple 3D encoder-decoder modules proved to be effective at improving accuracy. In order to keep computational costs manageable, features are extracted down to quarter resolution before building the cost volume, thus leading to about twice as fast inference compared to GC-Net [73].

EMCUA [77]. Other authors worked on expanding the contextual information processed by neural networks in the early stages. Nie *et al.* introduce *Multi-level Context Ultra-Aggregation* (MCUA). Given a branch working at a certain resolution (*i.e.* half) in a dense network, a child module, sharing weights with the main branch, is deployed to process a downsampled version of the same features (*i.e.* at quarter resolution). Usually, the output of the main branch reaches the same lower resolution and is processed by a new branch. At this point, features extracted by each layer of the child module are densely connected to this new branch, actually implementing inter-level interactions.

CSPN [78] To further improve PSMNet performance, Cheng *et al.* [78] propose Convolutional Spatial Propagation Network (CSPN) modules, capable of learning an affinity matrix for feature aggregation and spatial propagation of 2D unary features. By extending CSPN design to 3D, information is also propagated within the disparity dimension, enabling aggregation over both spatial and cost dimensions when processing features from 3D encoder-decoders.

GA-Net [79]. Matching cost aggregation is crucial in conventional methods, where local aggregation techniques [80], [81] or semi-global optimization [26] are widely adopted. Cost aggregation is beneficial even in deep neural networks, as demonstrated by the efforts to improve the design of encoder-decoder modules. Zhang *et al.* propose two novel layers, aimed at capturing local and global cost relationships. They are a locally guided aggregation layer and a semi-global aggregation layer, respectively implementing a traditional cost filtering strategy and a differentiable approximation of the SGM algorithm. By replacing 3D convolutions with few instances of these layers, their *Guided Aggregation network* (GA-Net) easily outperforms models deploying dozens of traditional, costly convolutions.

StereoDRNet [82]. Although accurate, deep stereo networks often produce geometrically inconsistent disparity maps, which negatively affect higher-level applications such as 3D reconstruction via *Truncated Surface Distance Function* (TSDF) fusion. This leads Chabra *et al.* to improve PSMNet design [75] to obtain more geometrically consistent predictions and thus better 3D reconstructions by fusing them. Their contributions include the use of a Vortex pooling layer [83] which proved to be more effective compared to the SPP layer, the introduction of 3D dilated convolutions inside the stacked encoder-decoders, and a refinement network for enhancing the initial disparity map.

PDSNet [84]. Most deep networks are memory-hungry and have to be trained for a given target disparity range. Tulyakov *et al.* propose *Practical Deep Stereo Network* (PDSNet) to address both limitations. They decrease the memory footprint by introducing a bottleneck matching module, which compresses the concatenated features from the two images into compact matching signatures, processed by a 3D encoder-decoder network to infer a sub-pixel MAP approximation. Thus, a weighted mean is computed around the disparity with the maximum posterior, which is robust to erroneous modes in the disparity distribution and allows to modify the disparity range without re-training. A novel sub-pixel criterion, derived by combining the standard cross-entropy loss with kernel interpolation, leads to faster convergence rates and higher accuracy.

StereoNet [85]. A weakness of 3D architectures compared to 2D models is the computational effort required even for a single inference. On average, 3D networks have about one order of magnitude higher memory requirements and runtime because of the additional dimension. The most expensive operations are those performed at the highest resolutions, thus Khamis *et al.* design a 3D model limited to a low-resolution volume (*i.e.* $\frac{1}{8}$), from which a coarse disparity map is extracted. This latter is sequentially upsampled and refined through shallow 2D networks. Thanks to the much lower complexity of 2D convolutions, StereoNet achieves much higher throughput than 3D networks at the cost of a marginal accuracy drop.

AnyNet [86]. Coarse-to-fine strategies that proved to be successful for 2D architectures have also been proposed for 3D networks. Concurrently to works on 2D networks [59], [60], Wang *et al.* deploy a pyramidal model that extracts a small number of feature maps from the images and then builds a very compact 4D volume by computing the L_1 distance between left and (warped) right features. The network works at three scales, deploying a coarse to fine disparity estimation strategy. After the last prediction, a Spatial Propagation network (SPNet [87]) produces the final output. The authors also endow their AnyNet model with an early-stopping functionality at *anytime*, *i.e.* inference can be shortened to obtain one of the coarser disparity maps, allowing for speed-accuracy trade-offs, as in the case of real-world applications with limited resources.

HSM [88]. High-resolution images have always been challenging for stereo matching, in particular in terms of resource requirements. As we observed so far, a standard strategy is to reduce the target resolution and rely on upsampling to restore it after inference. High-resolution stereo matching has been tackled by Yang *et al.* with the *Hierarchical Stereo* Matching (HSM) network. Following the pyramidal approaches discussed above, they extract a set of features at different resolutions and compute cost volumes with different search ranges according to the resolution. Each volume is processed to obtain a disparity map, upsampled to be concatenated with higher resolution volumes, and processed to produce finer disparity maps. For training, the authors propose a new, high resolution set of images (about 2056×2464) combined with available high [8] and low-

Method	Family	KITTI 2015 [9]			
		D1-bg% (↓)	D1-fg% (↓)	D1-all% (↓)	time (s)
CSPN [78]	3D	1.51	2.88	1.74	1.0
GA-Net [79]	3D	1.48	3.46	1.81	1.8
HD^3-Stereo [59]	2D	1.70	3.63	2.02	0.14
EMCUA [77]	3D	1.66	4.27	2.09	0.90
GWC-Net [89]	3D	1.74	3.93	2.11	0.32
SSPCV-Net [90]	3D	1.75	3.89	2.11	0.9
HSM [88]	3D	1.80	3.85	2.14	0.14
DeepPruner [91]	3D	1.87	3.56	2.15	0.18
DispNet-CSS [16]	2D	1.92	3.32	2.16	0.25
AutoDispNet-CSS [17]	2D	1.94	3.37	2.18	0.90
SENSE [71]	2D	2.07	3.01	2.22	0.32
SegStereo [66]	2D	1.88	4.07	2.25	0.60
StereoDRNet [82]	3D	1.72	4.95	2.26	0.23
PSMNet [75]	3D	1.86	4.62	2.32	0.41
ECA [74]	3D	2.14	3.45	2.36	0.22
iResNet [64]	2D	2.25	3.40	2.44	0.27
PDSNet [84]	3D	2.29	4.05	2.58	0.50
EdgeStereo [68]	2D	2.27	4.18	2.59	0.27
CRL [62]	2D	2.48	3.59	2.67	0.47
GC-Net [73]	3D	2.21	6.16	2.87	0.90
CNN+CRF [61]	2D+CRF	-	-	3.61	1.3
MC-CNN-acrt [27]	-	2.89	8.88	3.89	67
DispNet-C [11]	2D	4.32	4.41	4.34	0.06
MADNet [60]	2D	3.75	9.20	4.66	0.02
StereoNet [85]	3D	4.30	7.45	4.83	0.02
OASM-Net [92]	3D	6.89	19.42	8.98	0.73
OpenCV-SGBM [26]	-	8.92	20.59	10.86	1.1

TABLE 3
KITTI 2015 leaderboard [9], showing end-to-end methods.

Method	Family	Middlebury 2014 [8]		ETH3D [10]	
		Res.	bad 2.0%(↓)	avg. px (↓)	bad 1.0% (↓)	avg. px (↓)
MC-CNN-acrt [27]	-	H	8.08	3.82	-	-
HSM [88]	3D	F	10.2	2.07	4.00	0.28
CNN+CRF [61]	2D+CRF	H	12.5	-	-	-
SGM [26]	-	H	18.4	5.32	10.08	0.50
EdgeStereo [68]	2D	F	18.7	2.68	-	-
DispNet-CSS [16]	2D	H	22.8	4.04	2.69	0.22
iResNet [64]	2D	H	22.9	3.31	3.68	0.24
StereoDRNet [82]	3D	-	-	-	4.46	0.27
DeepPruner [91]	3D	Q	30.1	4.80	3.52	0.26
PSMNet [75]	3D	Q	42.1	6.68	5.02	0.33

TABLE 4
Middlebury 2014 [8] and ETH3D [10] leaderboards, showing end-to-end methods.

resolution [7], [9], [10] datasets. Moreover, they discuss the possibility of anytime on-demand inference, as in [86].

GWC-Net [89]. Various approaches for building volumes for 3D networks have been proposed, including feature concatenation, L_1 or L_2 distance. Guo *et al.* propose a new operator, namely *Group Wise Correlation layer* (GWC), placed in between feature concatenation and vector correlation. By treating the F unary features as N groups of structured vectors, the group-wise correlation layer computes N correlation scores, producing as output a new feature vector of dimension N. The fact that N is strictly smaller than F by definition reduces the computational efforts required by the first 3D convolutions, which are the most expensive ones due to the higher resolution. Moreover, this scheme provides a better feature representation enabling the network to infer more accurate disparity maps.

DeepPruner [91]. Another strategy for reducing the computational burden in 3D networks is to compute the matching costs for only a subset of all possible disparity hypotheses. To this aim, Duggal *et al.* deploy the PatchMatch algorithm [93] unrolled as a recurrent neural network to iteratively select random candidates, propagate them locally, and keep only the most promising ones. Moreover, a confidence score, which is inversely proportional to the range between the minimum and maximum disparity candidates, can be obtained for each pixel.

SSPCV-Net [90]. As proved by 2D networks, [66], [70], joint reasoning about disparity and semantics is beneficial to both. Along these lines, Wu *et al.* design a 3D network to pursue both tasks by leveraging on pyramidal cost volumes that are subsequently summed up at each resolution from coarse to fine. Moreover, from features relevant to segmentation a semantic cost volume is built and combined with the spatial volume to further include semantic information during disparity regression.

OASM-Net [92]. Although unsupervised learning proved to be effective thanks to image reprojection, its major shortcoming is at occlusions, where the reprojection fails to provide meaningful supervision. Li *et al.* address this problem by training a 3D network to jointly infer disparity and occlusions from a stereo pair. In addition to traditional reprojection losses [94], a regularization term allows for effective segmentation of occlusions, in order to properly take advantage of this information when computing the reprojection signals.

4.4 Experimental comparison

In this section, we report a quantitative comparison between the end-to-end architectures discussed so far. At first, we point out that each framework was trained according to different protocols (*e.g.* number of iterations, learning rate schedules, etc.). Thus, an entirely fair comparison is not possible without using the same protocol for each network. Nonetheless, we believe that we can fairly compare networks based on their top-performing configuration reported on the leaderboards.

KITTI 2015 [9]. The KITTI dataset is the preferred benchmark for deep stereo models, thanks to the low-variability image content of the driving scenario and a large number of training samples available for fine-tuning. This fact becomes apparent after observing the number of deep learning entries in the leaderboard: at the time of writing more than 100 top entries make reference to deep architectures, published or unpublished. Indeed, all the methods discussed in Section 4.2 and Section 4.3 are available on the online benchmark, with only a few exceptions [70], [72], [86], [92]. Table 3 collects them for a direct comparison, reporting also results for SGM and MC-CNN-acrt as baselines (highlighted in yellow). As in Section 3.4, we show D1-bg, D1-fg, and D1-all metrics together with the runtime reported by the authors, although measured on different hardware.

We observe that faster models (DispNet-C, MADNet and StereoNet) aiming for real-time performance achieve the worse D1-all score among all methods including the MC-CNN-acrt pipeline. We also point out that unsupervised models like OASM-Net already outperform conventional non-data-driven algorithms like SGM. The first end-to-end method outperforming MC-CNN-acrt is by Knöbelreiter *et al.* [61], combining a CNN and a CRF. Nevertheless, almost all end-to-end CNNs outperform this hybrid strategy. Climbing the leaderboard, we can observe how most of 3D architectures consistently outperform 2D stereo networks, with few exceptions [59], hinting that the former family excels at modeling geometry . Nevertheless, we also observe that early 3D models were particularly slow, *e.g.* GC-Net

[73], while the latest methods from this category are much faster, as in the case of GWC-Net [89]. CSPN [78] is the top-performing, published model on KITTI. Overall, we believe that HSM and HD^3 represent the best trade-off between accuracy and speed.

Middlebury 2014 [8]. Even though end-to-end models excel on the KITTI benchmark yielding small error rates, we are far from considering stereo as a solved problem. In particular, as clearly demonstrated by the Middlebury 2014 benchmark, high-resolution images depicting heterogeneous indoor environments and the meager amount of available training samples pose a major challenge for most neural networks. Therefore, this dataset is less popular for evaluating new architectures, indeed only five of the frameworks discussed in this section appear in the leaderboard, and they are not near the top.

Table 4 collects these results. We can notice how, in general, the percentage of bad pixels is much higher compared to KITTI, with average pixel errors higher than the prefixed threshold. The much higher resolution ($\sim 20\times$ compared to KITTI) and complexity of the scenes play a crucial role in making this dataset much more challenging for end-to-end models. Moreover, most methods cannot afford to process full resolution images during inference, with EdgeStereo and HSM the only models somehow dealing with them. In particular, the latter achieves the best results among all architectures, confirming the effectiveness of the hierarchical approach for high-resolution images. Nevertheless, it ranks only 31^{st} on the online leaderboard. This fact highlights that end-to-end models are still far from being state-of-the-art for stereo in any environment: indeed, the MC-CNN-acrt pipeline [27] outperforms most end-to-end models, except HSM on average error. This emphasizes how future research should focus on the development of robust networks better generalizing over content and resolution.

ETH3D [10]. Table 4 also collects the submissions to the ETH3D online benchmark by end-to-end approaches. Despite the limited training samples available, given the less heterogeneous image content and the much lower image resolution compared to the Middlebury 2014 dataset, the error rates achieved by the five architectures appearing in the leaderboard are much lower, with DispNet-CSS being the top-performing method among all submissions to the benchmark. This outcome confirms how most of the open challenges for deep stereo models are linked to high-resolution images together with complexity and variety of the observed environment.

5 Confidence Estimation

Almost in parallel to the first attempts of replacing single steps in the stereo pipeline, learning-based confidence estimation [96], aiming to predict reliability of the disparity assigned to each pixel, gained popularity. The first approaches relied on random forest classifiers [38], [39], [41], [42], [43], [44], [97] fed with conventional (*i.e.* not learning-based) features [98], while the most recent ones on CNNs [45], [99], [100]. Confidence estimators are typically trained on the output of a stereo algorithm using a two-class (inlier, outlier) label for each pixel, obtained from ground truth depth data by setting a threshold to distinguish between inliers and outliers in the output of the considered stereo algorithm. Moreover, techniques for self-supervised training of confidence estimators from video sequences [101] or traditional confidence measures [102] have been proposed in the literature.

Starting from the recent review and evaluation reported in [96], we introduce and classify novel approaches which have appeared since then. Specifically, we divide these techniques into two categories: those operating in the disparity/image domain and those processing the cost volume. Fig. 4 shows an example of a disparity map and the associated confidence map.

5.1 Confidence estimation in disparity/image domain

This family of confidence estimators directly learn the reliability of each pixel from the disparity map, and optionally the reference image. Although processing limited cues, these measures are particularly appealing when the full cost volume is not available, *e.g.* when using off-the-shelf stereo or end-to-end models not having a cost volume at all.

LFN [103]. Fu *et al.* propose to extend patch-based strategies for confidence estimation from the disparity map [45], [99], [100] in a Late-Fusion Network (LFN), which combines features extracted from both the image and disparity map, and introduce dilated convolutions to further increase the local context and potentially give more cues to the network for estimating confidence.

LGC-Net [95]. Tosi *et al.* extend the previous approach by considering both local cues (encoded by patches) and the global context by designing a *Local Global Confidence Network* (LGC-Net), combining the large receptive field of a global sub-network with the accuracy on high-frequency noise enabled by patch-based strategies.

5.2 Confidence estimation using the cost volume

Although the cost volume is often hidden from the end-user, it provides additional cues that are meaningful to distinguish reliable disparities from unreliable ones. For instance, it can encode the presence of multiple cost hypotheses competing for the minimum, which is information that cannot be retrieved from the disparity map alone.

Reflective confidence [48]. Following the trend of replacing single steps of the stereo pipeline, Shaked and Wolf propose to jointly estimate a confidence measure together with cost optimization, before disparity selection. A two-layer fully connected network processes the matching costs, predicting confidence together with the final disparity map.

Feature Augmentation [104]. As CNN based measures showed great results processing local cues from the disparity map, Kim *et al.* apply the same principle to confidence estimation based on random forests, by extracting a robust set of features extracted from super-pixel and concatenated with per-pixel features, similarly to [105].

Unified Network [106]. Jointly learning cost optimization and confidence estimation by working on small patches proved to be effective at improving the accuracy of the final disparity map of a stereo pipeline. Thus, Kim *et al.* propose a unified architecture for cost volume optimization and confidence estimation. A first encoder-decoder module refines the matching costs with a large receptive field in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3070917, IEEE Transactions on Pattern Analysis and Machine Intelligence

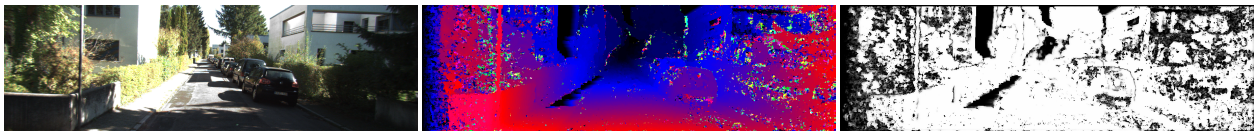

Fig. 4. **Example of confidence estimation.** From left to right, reference image from KITTI 2012 dataset, disparity map by MC-CNN-fst [27] raw algorithm and confidence estimation inferred by LGC-Net [95].

order to obtain a more accurate disparity map. Then, a final sub-network processes it together with top-k refined costs to output a confidence map.

LAF-Net [107]. A larger receptive field is usually effective in improving confidence estimation [95], [106]. The size of the receptive field of a neural network is traditionally determined by its architecture, *i.e.* the number of down-sampling operations performed. Kim *et al.* develop a novel model that extracts features from the image, disparity map and cost volume to infer confidence scores. A key element of this architecture is the scale inference network, which learns the scale map and warps the fused confidence features through convolutional spatial transformer networks [108].

5.3 Experimental comparison

In this section, we report a quantitative comparison between confidence estimation frameworks. We refer to experiments reported in [107], since it is the most recent publication that includes a fair comparison of relevant methods. All methods have been re-trained by the authors [107] on MPI Sintel [15] and KITTI 2012 [7]. The Area Under the Curve (AUC) metric [96], [98] over sparsification curves is used to evaluate the effectiveness of confidence measures with the error threshold set to 1.

Table 5 reports results on KITTI 2015 and the half-resolution images of Middlebury 2014. We also report the optimal AUC, highlighted in yellow. Confidence estimation is carried out on disparity maps generated by the regular Semi-Global Matching (SGM) algorithm [26] with a census-based matching cost function and the MC-CNN-acrt pipeline [27], including SGM optimization and filtering. The first three rows report results of top 3 methods evaluated in [96], namely O1, PBCP, CCNN. The table shows how increasing the size of the receptive fields improves confidence estimation. Indeed LGC-Net and LAF-Net are the top-performing methods on both datasets. Moreover, considering cost volume information together with image and disparity cues leads to minor improvements, enabling LAF-Net to achieve the best overall results. Nonetheless, it is worth pointing out that less constrained methods based only on image/disparity cues, such as LGC-Net, achieve very competitive results.

This evaluation, together with those reported in previous works [96], [98], highlights how confidence measures are very close to optimal performance when dealing with conventional stereo algorithms, even if they include learning-based modules. However, the literature lacks papers studying confidence estimation in the case of end-to-end stereo networks. Although there are published approaches [16], [109], [110], [111], [112] for estimating the uncertainty of CNNs, they have not been applied in this specific field so far, making it an exciting future research direction.

Method	KITTI 2015		Middlebury 2014	
	SGM	MC-CNN-fst	SGM	MC-CNN-fst
O1 [43]	4.61	2.63	7.91	7.07
PBCP [45]	4.39	2.72	7.91	7.18
CCNN [99]	4.19	2.58	7.69	7.16
LFN [103]	4.05	2.53	7.52	6.92
LGC-Net [113]	3.92	2.36	7.35	6.85
Augment. [104]	4.30	2.94	7.72	7.01
Unified [106]	4.07	2.50	7.49	6.94
Reflective [48]	5.31	2.92	8.06	7.36
LAF-Net [107]	3.85	2.25	7.18	6.83
Optimal	3.48	1.70	5.69	5.27

TABLE 5
Experimental comparison of confidence estimators. AUC scores ($\times 10^2$) computed on KITTI 2015 and Middlebury 2014 datasets.

6 Domain-shift: Challenges and Solutions

We have seen that end-to-end models achieve state-of-the-art performance on most benchmarks [7], [9], [10] and yield promising results on more challenging datasets [8], as shown in Section 4.4. This is made possible by training deep models on a large enough number of synthetic images to allow such complex architectures to converge. However, although thousands of images with different content can be easily obtained using computer graphic techniques, they currently fail at modeling many challenging properties of real imagery. In particular, the inability to accurately model camera noise, poor illumination conditions, reflections or brightness saturation produce notable loss of accuracy in disparity estimation on real stereo pairs due to the domain shift faced by the neural network. Fig. 5 depicts an example of this phenomenon, showing the disparity maps predicted by the same model, GWC-Net [89], when trained on synthetic images only or fine-tuned on real images from the target domain.

Although theoretically possible, it is practically infeasible to collect enough images with ground truth depth for all possible real environments. To overcome this limitation, three main categories of techniques aiming to bridge the gap between synthetic and real domains have been proposed: i) image synthesis and domain transfer, ii) self-supervised adaptation and iii) guided deep-learning. The first category comprises methods that learn a mapping function across domains [114], [115], in order to make synthetic images more realistic for fine-tuning or testing. These techniques are general and can be applied seamlessly to different tasks; thus are beyond the scope of this survey. Therefore, we will focus on the remaining two, in Section 6.1 and Section 6.2, respectively. Since different authors have deployed very different evaluation protocols, we refer to the papers for the experimental validation of each method.

6.1 Self-supervised adaptation

Conversely to other tasks for which, although tedious, manual annotation is feasible in an offline manner (*e.g.* semantic

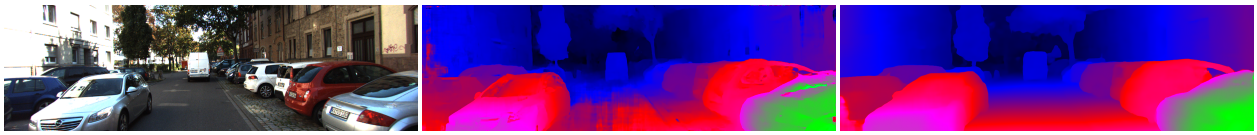

Fig. 5. **Effects of domain-shift.** On a KITTI 2015 stereo pair (top), a GWC-Net [89] instance trained on synthetic images [11] produces poor results (middle) on the road and in reflective surfaces. A short fine-tuning on KITTI 2012 dramatically improves the results (bottom).

segmentation or object detection), accurately labeling images with depth measurements for training requires expensive active sensors, a cumbersome setup with appropriate calibration and often further offline refinement.

On the other hand, acquiring unlabelled image pairs is much simpler since it only requires calibrated stereo cameras. This makes the possibility of *self-adapting* a neural network in the absence of ground truth labels particularly appealing, since it dramatically reduces the efforts required to bridge the gap across domains. We distinguish two main paradigms: *offline* and *online* – in case adaptation is carried out before encountering the new environment or directly during deployment in parallel with inference.

6.1.1 Offline adaptation

We classify into this category techniques that adapt a pre-trained neural network to a new domain before its deployment. These strategies are similar to conventional fine-tuning, but do not need ground truth labels.

Confidence-guided Adaptation [116], [117]. Although traditional algorithms such as SGM [26] are widely outperformed by well-trained neural networks, they do not suffer from large drops in accuracy when applied in different domains. Moreover, confidence measures [96] proved to be particularly effective at detecting outliers, making it possible to filter out significant errors and only keep sparse, but accurate disparity labels. Driven by this rationale, Tonioni *et al.* propose to leverage traditional (*i.e.* not learning-based) stereo algorithms, such as SGM [26] or *block matching*, and a confidence measure, specifically CCNN [99], to process a set of unlabelled stereo pairs in order to retrieve a set of sparse, yet reliable disparity maps. Such sparse depth measurements are then used as *proxy labels* for fine-tuning a deep network pre-trained on synthetic images only.

Self-supervised CNN-CRF [118]. Following a similar rationale, Knöbelreiter propose to improve the performance of the CNN-CRF framework [61] to adapt to different image domains, e.g. to aerial images. In particular, they carry out a self-supervised training of the CNN in charge of computing the data term, using as labels the output of the full CNN-CRF filtered by a left-right consistency check.

Zoom and Learn [119]. Pang *et al.* observe that processing higher resolution images yields more detailed disparity maps and leverage this fact to improve the performance of models trained on different domains. In their *Zoom and Learn* framework, a neural network is presented with a stereo pair and an upsampled version of the same images, thus training the model to reproduce the results achieved for the higher resolution images when lower resolution images are processed. As a result, the model learns to infer finer details from the images and thus to better tackle difficulties in generalization across domains.

6.1.2 Online adaptation

This paradigm removes the division between the training and testing phases. It *continuously trains* the model any time new data are available, overcoming the need for data from the new domain before deployment. On the other hand, adaptation starts at deployment, making early results inaccurate, but gradually improving over time.

Open World Stereo [120]. Inspired by the possibility of learning depth estimation through view synthesis [94], Zhong *et al.* develop *Open World Stereo*, which is a 3D convolutional LSTM capable of fast convergence, typical of the 3D networks and reinforced by the memory mechanism introduced by LSTM modules. After an initial, *prime* training phase, Open World Stereo is deployed on a new environment and, for each newly observed stereo pair, estimates an output disparity map. This latter is used at the same time to get supervision signals by warping the right image towards the left and measuring the appearance error with respect to the left image. Although a single inference takes more than a second, the network rapidly adapts to the new environment in a few hundred iterations.

Real-Time Self-Adaptive deep stereo [60]. The possibility of adapting *on-the-fly* to a new environment is particularly appealing to make a system truly portable. Tonioni *et al.* propose the first framework for real-time online self-adaptation. It relies on the synergy of two main components: i) a fast, Modularly Adaptive Network (MADNet), already introduced in Section 4.2, and ii) an effective strategy for adapting only different portions of the entire network. For each new incoming stereo pair, a portion is chosen according to a heuristic and supervision signals are computed through warping with the disparity estimated at the chosen resolution. Finally, back-propagation is performed only in layers belonging to the selected portion of the network. This strategy enables very fast back-propagation but, on the other hand, increases the number of steps required to converge compared to back-propagating over the entire network. The result is an approximation over time of full back-propagation that maintains real-time inference.

Learning to adapt [19]. Efficiency, by maximizing accuracy improvement out of each adaptation step, is desirable when adapting online to new environments. To achieve a starting parameter configuration that is suitable for adaptation, Tonioni *et al.* propose the *Learning to Adapt* (L2A) training protocol. By incorporating the adaptation process itself into the learning objective through *meta-learning*, the network is predisposed to a configuration of parameters that are better suited for online adaptation. This means that each offline training iteration mimics online adaptation steps over a small, synthetic sequence with ground truth. The performance gain of the adaptation phase is used as supervision for the network, leading to more efficient opti-

mization steps.

6.2 Guided deep learning.

This paradigm differs from previous adaptation techniques because it aims at mitigating difficulties due to domain shift without requiring explicit on-the-fly fine-tuning. The rationale behind this approach is to *guide* a network by providing external *hints*. For instance, given sparse depth information obtained by any means (*e.g.* LiDAR sensors) this method aims at overcoming the biases of a network by driving it to the correct depth values. The idea is that, since the domain shift issue is due to the appearance gap between domains (*e.g.* synthetic vs. real images), the depth cues can overcome the loss of accuracy due to domain changes.

Poggi *et al.* [121] propose a strategy that enhances or dampens feature activations in a neural network to modulate its predictions. This is accomplished by centering a Gaussian function on each depth value provided by the sparse cues and modulating features that have a strong relationship with the output according to the Gaussian function. These features come, respectively for 2D and 3D networks, from correlation scores or features concatenation/difference, as discussed in Section 4.2 and Section 4.3. This technique can be applied during both training and testing to greatly reduce the effects of domain shifts.

7 Monocular depth estimation through stereo supervision

In this section, we move our focus to a new and exciting research trend: depth estimation from a single image, for which the synergy between stereo and deep learning recently allowed for results unimaginable just a few years ago. In monocular depth estimation, the goal is to learn a non-linear mapping between a single RGB image and its corresponding depth map. Even though this task comes natural to humans, it is an ill-posed problem, since a single 2D image might originate from an infinite number of different 3D scenes. However, unlike multi-view setups, a single-image depth estimation system does not require any additional equipment, making it applicable in countless scenarios. Early supervised learning approaches [122], [123], [124], [125], [126] have been quite successful in this task. However, such models typically require vast amounts of pixel-wise, ground truth training annotations which are very difficult to obtain, and thus suffer from the limitations of end-to-end stereo approaches analyzed in Section 4.1.

More recent self-supervised strategies cast depth estimation as a view-synthesis problem by introducing a photometric reconstruction loss to avoid the need for expensive ground truth depth annotations. These methods received a lot of attention since they were able to take advantage of existing, or easy-to-collect, large datasets comprising either stereo pairs or monocular videos. As opposed to multi-frame visual odometry, approaches based on stereo images do not suffer from scale ambiguity due to the known camera baseline. Based on this, inferring depth without scale ambiguity results feasible even using a single RGB image as input, given the same camera at training and test time. This highlights once more the synergy between stereo and deep learning when dealing with depth estimation. In addition, with the evolution of this research trend more supervisory signals have been introduced, ranging from clever usage of the image reprojection principle to the adoption of distillation paradigms aimed at obtaining stronger loss terms. These techniques have greatly shrunk the gap with supervised approaches in terms of accuracy.

Geometry to the Rescue [3]. This work is the first to estimate depth from a single image using two images of a stereo pair, referred to as source I_L and target I_R, for training. A coarse-to-fine end-to-end convolutional neural network is trained to perform novel view synthesis, minimizing the photometric difference between the input image I_S and a reconstructed one I_W. The proposed architecture infers scaled inverse depth (*i.e.* disparity) from the source image I_S, which is then used to synthesize the target image I_w adhering to epipolar constraints.

MonoDepth [127]. Other researchers follow the above seminal approach with a wide range of solutions and technical contributions. MonoDepth is an encoder-decoder architecture performing single image depth estimation in a self-supervised manner. Its main characteristics include i) a new training loss enforcing consistency between the predicted inverse depth maps aligned with each camera view, ii) the use of a fully sub-differentiable training loss based on the existing bilinear sampling strategy [108], iii) a robust appearance reconstruction loss, combination of an $L1$ term and a simplified single scale SSIM [128] loss that compares back-warped images with their real counterparts and iv) a post-processing step to soften artifacts near occlusions, requiring two forward passes at test time.

AsiANet [129]. Most deep networks share the same architectural design, consisting of an encoder-decoder structure based on U-Net and various encoders, with VGG and ResNet being the most popular. In contrast, Yusiong *et al.* develop a framework dubbed *Autoencoders in Autoencoders network* (AsiANet) by stacking multiple autoencoders in a multi-scale setting. Specifically, the authors employ a unique Inception-like pooling module, based on fractional max-pooling in the encoding part and multi-scale cascaded autoencoders in the decoder. This design benefits from multi-scale features when upsampling the output of the encoder taking into account local and global cues.

3Net [130]. Despite the surprising effectiveness of stereo supervision, it cannot handle occlusions during training, leading monocular networks to generate artifacts in these regions. To overcome this, Poggi *et al.* design 3Net and supervise it by assuming three horizontally aligned images, learning to estimate two depth maps for the central frame supervised respectively by the remaining two. These two maps show occlusions in specular regions that are compensated by combining them. Due to the lack of trinocular datasets, 3Net employs an interleaved training strategy and a custom architecture, designed to simulate a trinocular setting based on conventional binocular inputs.

SuperDepth [138]. Most of the architectures for monocular depth estimation described so far are trained using low-resolution images, due to memory constraints, and employ photometric losses. As a consequence, this forced design choice limits the attainable depth accuracy. To address this, Pillai *et al.* introduce a deep neural network relying on

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3070917, IEEE Transactions on Pattern Analysis and Machine Intelligence

Method	S	V	P	A	GT	F	CS	E2E	Res	Abs Rel (↓)	Sq Rel (↓)	RMSE (↓)	RMSE log (↓)	$\delta < 1.25$ (↑)	$\delta < 1.25^2$ (↑)	$\delta < 1.25^3$ (↑)
Eigen split [131] - 697 images (maximum depth: 80m)																
SfMLearner [132]		✓					✓	✓	416 × 128	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Monodepth2 [133] †		✓						✓	1024 × 320	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Geo. to the Rescue [3]	✓							✓	608 × 176	0.152	1.226	5.849	0.246	0.784	0.921	0.967
Monodepth-R50 [127]	✓						✓	✓	512 × 256	0.114	0.898	4.935	0.206	0.861	0.949	0.976
AsiANet [129]	✓						✓	✓	512 × 256	0.128	1.161	5.470	0.213	0.858	0.947	0.974
3Net-R50 [130]	✓						✓	✓	512 × 256	0.111	0.849	4.822	0.202	0.865	0.952	0.978
MonoGAN-VGG [134]	✓						✓	✓	512 × 256	0.118	0.908	4.978	0.210	0.855	0.948	0.976
CRF-DGAN [135]	✓							✓	512 × 256	0.135	1.182	5.582	0.235	0.828	0.933	0.967
StrAT [136]	✓							✓	512 × 256	0.128	1.019	5.403	0.227	0.827	0.935	0.971
PyD-Net [137]	✓						✓	✓	512 × 256	0.146	1.291	5.907	0.245	0.801	0.926	0.967
SuperDepth [138]	✓							✓	1024 × 384	0.112	0.875	4.958	0.207	0.852	0.947	0.977
Refine and Distill [139]	✓								512 × 256	0.098	0.831	4.656	0.202	0.882	0.948	0.973
Monodepth2 [133] †	✓	✓						✓	1024 × 320	0.106	0.806	4.630	0.193	0.876	0.958	0.980
Zhan [140]	✓	✓						✓	608 × 160	0.135	1.132	5.585	0.229	0.820	0.933	0.971
EPC++ [141]	✓								832 × 256	0.127	0.936	5.008	0.209	0.841	0.946	0.979
DVSO [142]	✓	✓	✓						512 × 256	0.097	0.734	4.442	0.187	0.888	0.958	0.980
monoResMatch [113]	✓		✓				✓	✓	1280 × 384	0.096	0.673	4.351	0.184	0.890	0.961	0.981
Depth-Hints [143] †	✓		✓					✓	1024 × 320	0.096	0.710	4.393	0.185	0.890	0.962	0.981
Kuznietsov [144] †	✓				✓			✓	621 × 187	0.113	0.741	4.621	0.189	0.862	0.960	0.986
SVS [145]	✓				✓	✓			640 × 192	0.094	0.626	4.252	0.177	0.891	0.965	0.984
Cross-domain [146]	✓		✓		✓	✓		✓	512 × 256	0.096	0.641	4.095	0.168	0.892	0.967	0.986
KITTI 2015 split - 40 images																
Monodepth-R50 [127]	✓						✓	✓	512 × 256	0.159	2.411	6.822	0.239	0.830	0.930	0.967
SemMonodepth-R50 [147]	✓			✓			✓	✓	512 × 256	0.143	2.161	6.526	0.222	0.850	0.939	0.972
KITTI odometry split - 8691 images																
Monodepth-R50 [127]	✓						✓	✓	512 × 256	0.108	0.679	4.123	0.194	0.868	0.952	0.978
VOMonodepth-R50 [148]	✓	✓	✓				✓	✓	512 × 256	0.091	0.548	3.690	0.181	0.892	0.956	0.979

TABLE 6
Quantitative evaluation on the KITTI dataset [12], Eigen test split [131]. †indicates feature extractors pre-trained on ImageNet [149]. S: Stereo, V: Video, P: Proxy, A: Additional information, GT: Ground Truth, F: Freiburg SceneFlow, CS: Cityscapes, E2E: End-to-End, Res: Resolution.

techniques typically used for super-resolution. In particular, they show that by simply increasing the input image resolution, depth estimation significantly improves accordingly. Then, taking as reference the MonoDepth [94] architecture, SuperDepth obtains much better results by incorporating *sub-pixel convolutional layers* to *super-resolve* inverse depth maps at each scale (typically up-sampled by means of standard *deconvolutions* or *resize-convolution* sequences) and a *differentiable flip-augmentation layer*, in order to mitigate occlusion artifacts in an end-to-end fashion.

SVS [145]. Although it is not a self-supervised methodology, SVS is the very first attempt to mimic stereo matching for monocular depth estimation. To this aim, processing occurs in two stages exploiting two distinct architectures. A first view synthesis network, based on Deep3D [150], is in charge of generating a synthetic right view selecting pixels from the input image I_L. Then, a second depth-supervised network based on the DispNet-C structure [11] processes the left and the synthesized right images to estimate the final depth map, achieving state-of-the-art results.

Cross-domain [146]. Although affected by the domain-shift curse, as discussed in Section 6, deep stereo networks are in general more accurate than monocular ones across domains because they reason about geometry. Following this rationale, Guo *et al.* generate distilled depth labels, employing stereo networks trained on large synthetic datasets with ground truth and optionally fine-tuned on realistic ones. Such distilled knowledge provides dense supervision for a monocular network, resulting much more effective than warping strategies.

MonoResMatch [113]. Although distilling knowledge from stereo networks shows promise [146], it requires a two-stage training protocol and is affected by domain-shifts. To overcome both, Tosi *et al.* propose respectively i) a novel architecture, jointly estimating a virtual view and performing stereo matching between it and the real image, and ii) to leverage a traditional stereo algorithm agnostic to domain, such as SGM, for distillation. As consequence, no synthetic data is required at all and a single, end-to-end training is carried out to achieve state-of-the-art accuracy.

Refine and Distill [139]. Following distillation approaches, Pilzer *et al.* propose a framework for self-supervised depth estimation comprising two collaborative architectures. A *student network* is in charge of synthesizing I_W as opposite view to the input image (counter-intuitively, I_R in this framework). Then, a backward cycle network attempts to re-synthesize the original input image taking I_W as input. A *teacher network* takes advantage of the inconsistencies between input image and I_W to infer a refined depth map. The last step exploits knowledge distillation, in order to transfer information from *teacher* to *student* and thus, to improve the *student network*.

Depth-Hints [143]. By studying the training signals enabled by reprojection, Watson *et al.* show that finding the optimal depth value is often not trivial, especially in regions of the images where the photometric loss is ambiguous and multiple depth candidates appear valid. In order to alleviate these problems, they propose to rely on external *depth hints* obtained from *off-the-shelf* stereo algorithms. Depth hints are trusted only when showing to be more reliable than image reprojection, thus providing complementary information during the training phase.

MonoGAN [134]. With the advent of Generative Adversarial Networks (GANs), it became possible to model distributions of complex data, enabling new capabilities in deep image synthesis for instance. Aleotti *et al.* use a generator network based on MonoDepth to infer depth from I_L and synthesize the warped target image I_W. A discriminator network is then used to distinguish between *fake* I_W and *real* I_R. Thus, the discriminator pushes the generator to obtain more realistic I_W and, thus, better depth estimates.

StrAT [136]. Following the success of GANs, this work

addresses monocular depth estimation by proposing a novel *Structured Adversarial Training* (StrAT) strategy. Specifically, a generator model tries to generate realistic stereo pairs, which have to be discriminated from the real ones. By incrementally varying baselines, the authors show that multiple samples with varying degrees of difficulty can be generated to guide the training process of the entire architecture.

CRF-DGAN [135]. By studying in depth the previous GAN-based approaches, Puscas *et al.* propose a dual generative adversarial network, coupled with a structured Conditional Random Field (CRF), for depth prediction. In particular, two generative models infer different, complementary, disparity maps that are then fused and further processed with the outputs of two discriminative networks using the CRF. By doing so, the entire architecture establishes strong mutual constraints between each component in order to facilitate network optimization, and thus depth generation.

PyDNet [137]. Although previous works mostly focus on accuracy, real-time and low-power constraints are often also crucial in real applications. PyDNet represents the first attempt to make monocular depth estimation feasible on standard CPUs and embedded devices with limited memory capacity. This goal is achieved thanks to a modular design, based on a *pyramidal feature extractor* and a series of shallow *depth decoders*. The network infers depth maps from $\frac{1}{64}$ to half of the input resolution and allows for *early-stop* at intermediate resolutions, in case of strict timing constraints. PyDNet provides results comparable to standard networks [94] but considerably faster, achieving high fame rate for depth inference on embedded devices [137] and on low-power platforms [151].

Semi-Supervised Monocular Depth [144]. Arguing that typical depth sensors, such as 3D laser scanners, have specific noise characteristics and generate measurements much sparser than images, while self-supervised strategies based on stereo images struggle in texture-less regions, Kuznietsov *et al.* propose a semi-supervised methodology to incorporate the best of both worlds. Specifically, the network exploits 3D laser measurements in supervised fashion and, at the same time, stereo pairs using a direct image alignment loss based on photometric consistency.

Semantic MonoDepth [152]. Semantic segmentation has witnessed enormous progress due to machine learning. Therefore, learning to infer both semantics and depth from a single image exploiting the synergies of the two is of particular interest. For this purpose, Ramirez *et al.* unite self-supervised monocular depth estimation and supervised semantic segmentation by introducing i) a shared encoder and task-specific decoders trained for joint optimization and ii) a *cross-domain discontinuity loss* to enforce spatial proximity between depth discontinuities and semantic contours.

Deep Feature Reconstruction [140]. An alternative path consists into learning single image depth estimation from monocular sequences, captured by a moving camera and assuming the scene to be static [132]. In this case, the pose between frames is not known, thus estimated depth suffers from scale ambiguity. Zhan *et al.* propose to tackle this latter problem by using stereo sequences at training time. By imposing both spatial and temporal constraints, scene depth and camera motion are in a common real-world scale.

Monodepth2 [133]. Most of the research on monocular depth estimation focused on complex architectures or specific loss functions. Godard *et al.* show that careful design choices in conjunction with light-weight models suffice to obtain state-of-the-art results. Considering joint supervision from a monocular sequence and a stereo pair, the authors propose i) a minimum reprojection loss to effectively handle occlusions between consecutive frames, ii) a multi-scale photometric loss, upsampling low-resolution intermediate depth maps to full resolution for better supervision and iii) an auto-masking loss to ignore pixels that violate camera motion assumptions during the training phase.

EPC++ [141], [153]. More authors combined supervision from sequences with stereo setup. Luo *et al.* propose EPC++ (Every Pixel Counts++), taking as input two images of a monocular sequence and adopting three task-specific networks to predict camera motion, depth and optical flow. A further component, named holistic 3D motion parser (HMP), is in charge of recovering a per-pixel 3D motion for both rigid background and moving objects.

DVSO [142]. To reduce the limitations of traditional, monocular visual-odometry, which is typically prone to scale drift issues due to the unknown absolute metric scale, Yang *et al.* incorporate dense monocular depth prediction into a monocular odometry pipeline. To this aim, the authors design a novel monocular network architecture called *StackNet* and built by stacking two-subnetworks, respectively *SimpleNet*, that learns to infer an initial depth estimate in a first training phase, and *RefineNet*, that refines it in a second training phase. Photometric consistency losses are combined with proxy labels, sourced from a stereo odometry algorithm, in order to supervises the framework.

VOMonodepth [148]. Since monocular sequences are often available during deployment too, it seems reasonable to feed a monocular network with sparse 3D cues obtained with a traditional Visual Odometry (VO) algorithm. Andraghetti *et al.* feed a monocular network with *VO priors* to facilitate the training process and the accuracy for both complex and compact models. In contrast to strategies described so far leveraging visual odometry, it is the only solution for which external hints are available at test time as well.

7.1 Experimental comparison

We now report a comparison of the monocular networks reviewed so far. To this aim, we collect performance measured on the KITTI raw dataset [7], evaluating a set of metrics proposed by Eigen *et al.* [131]: four error metrics, the lower the better, respectively Absolute Relative error (Abs Rel), Squared Relative error (Sq Rel), Root Mean Squared Error (RMSE) and scale-invariant RMSE (RMSE log); and three accuracy scores, obtained by counting the fraction of total pixels for which δ, that is the maximum between the predicted depth to ground truth and the ground truth to depth ratios, is lower than 1.25, 1.25^2 and 1.25^3, the higher the better. Table 6 collects results of different models on different splits of KITTI, taken from the original papers. For each method, we indicate the kind of supervision (S: Stereo, V: Video sequence, P: Proxy labels, A: Additional, GT: Ground Truth), additional datasets used for training (F: Freiburg SceneFlow, CS: CityScapes), the resolution adopted at training/testing time (Res) and the capability of the network to be trained in an end-to-end manner (E2E).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3070917, IEEE Transactions on Pattern Analysis and Machine Intelligence

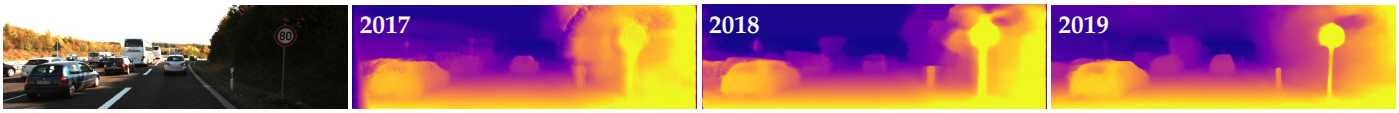

Fig. 6. **Evolution of stereo-supervised monocular depth estimation**, showing results achieved through 2017 [94], 2018 [130] and 2019 [113].

Most current approaches adopt the Eigen split [131] of KITTI, for which 697 images with "ground truth" depth acquired with a Velodyne sensor are used for testing and 22600 for training. Methods evaluated on different splits [147], [148] are reported on bottom and compared with their baselines [94]. Depth maps are evaluated within the first 80 meters using the Garg crop [3]. In this evaluation, we also consider methodologies trained only on monocular sequences [132], [133] in order to highlight the existing gap with stereo supervision and to point out how such a margin is progressively shrinking. We can observe how, in general, using proxy labels from stereo pairs or stereo sequences at training time as in [113], [139], [142] allows to notably improve the accuracy compared to other self-supervised strategies, obtaining comparable or better results than supervised and semi-supervised networks [144], [145], [146]. Moreover, image resolution and the pre-training process play a key role as well [113], [133], [138]. Finally, both semantic [147] and VO [148] priors are indeed beneficial to monocular depth estimation.

Figure 6 highlights the notable progress in monocular depth estimation, deploying stereo supervision, achieved in the past three years.

8 Discussion

In this section, we summarize the achievements of the recent methods reviewed in this paper and also identify the remaining open challenges and possible future research directions. We identify four main take-home messages:

- While seminal learning-based approaches aimed at replacing single steps of the stereo pipeline showed great potential, the greatest turning-point was due to the change from hand-crafted pipelines to end-to-end networks. This paradigm is nowadays dominant and represents the preferred choice for both experts and new researchers, whereas the popularity of hand-crafted pipelines is rapidly fading.
- Nevertheless, conventional knowledge about stereo survived this paradigm shift and has not gone extinct. Indeed, specific design choices such as the correlation layer [11] or 3D cost aggregation [73] are inspired by decades of research on stereo and play a key role in many deep networks.
- The main shortcomings introduced by end-to-end models concern the need for large amounts of ground truth annotated samples, that limits their seamless deployment in-the-wild. Self-supervised or adaptation techniques (Section 6.1) are emerging as promising strategies to address the problem, paving the way for brand new research opportunities.
- Stereo geometry turned out to also be a precious source of self-supervision for frameworks estimating depth from a single image [3], [94], by dramatically reducing the overhead required to collect training samples and rapidly contributing to the spread and development of this research thrust.

Nevertheless, two major challenges remain in this field: i) generalization across different domains and ii) applicability on high-resolution images. In particular, current results on Middlebury 2014 [8] highlight these open problems. Although a few works have started addressing the former by means of continuous adaptation [60] and the latter by careful design choices [88], in our opinion these will be the major directions of development in the upcoming years.

9 Conclusion

We have presented a comprehensive survey of recent advances in depth estimation from images leveraging the synergies between binocular stereo and data-driven, learning-based methods. Reviewing the literature reveals that the relationship is bidirectional and new machine learning approaches had to be developed to address depth estimation. While research is ongoing and voluminous, we believe that this survey will be valuable for researchers entering this field, as well as for experts.

References

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1-3, pp. 7–42, 2002.
[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[3] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *ECCV*. Springer, 2016, pp. 740–756.
[4] M. Brown, D. Burschka, and G. Hager, "Advances in computational stereo," *PAMI*, vol. 25, no. 8, pp. 993–1008, 2003.
[5] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *CVPR*, 2003.
[6] H. Hirschmüller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *CVPR*, 2007.
[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *CVPR*, 2012.
[8] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *German Conference on Pattern Recognition*, 2014, pp. 31–42.
[9] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *CVPR*, 2015, pp. 3061–3070.
[10] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *CVPR*, 2017, pp. 3260–3269.
[11] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *CVPR*, 2016, pp. 4040–4048.
[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *IJRR*, vol. 32, no. 11, pp. 1231–1237, 2013.

[13] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox, "What makes good synthetic training data for learning disparity and optical flow estimation?" *IJCV*, vol. 126, no. 9, pp. 942–960, 2018.

[14] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "ShapeNet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[15] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conf. on Computer Vision (ECCV)*, ser. Part IV, LNCS 7577, A. Fitzgibbon et al. (Eds.), Ed. Springer-Verlag, Oct. 2012, pp. 611–625.

[16] E. Ilg, T. Saikia, M. Keuper, and T. Brox, "Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation," in *ECCV*, 2018.

[17] T. Saikia, Y. Marrakchi, A. Zela, F. Hutter, and T. Brox, "Autodispnet: Improving disparity estimation with automl," in *ICCV*, 2019.

[18] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[19] A. Tonioni, O. Rahnama, T. Joy, L. Di Stefano, A. Thalaiyasingam, and P. Torr, "Learning to adapt for stereo," in *CVPR*, 2019.

[20] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *IJRR*, vol. 36, no. 1, pp. 3–15, 2017.

[21] P. Wang, X. Huang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape open dataset for autonomous driving and its application," *PAMI*, 2019.

[22] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou, "DrivingStereo: A large-scale dataset for stereo matching in autonomous driving scenarios," in *CVPR*, 2019.

[23] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," *arXiv preprint arXiv:1912.04838*, 2019.

[24] M.-F. Chang, J. W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *CVPR*, 2019.

[25] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet, "Lyft Level 5 AV Dataset 2019," url-https://level5.lyft.com/dataset/, 2019.

[26] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *PAMI*, vol. 30, no. 2, pp. 328–341, 2008.

[27] J. Žbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches." *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2287–2318, 2016.

[28] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *ECCV*, 1994, pp. 151–158.

[29] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," in *ICCV Workshops*, 2011, pp. 467–474.

[30] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 1073–1079, 2009.

[31] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang, "A deep visual correspondence embedding model for stereo matching costs," in *ICCV*, 2015, pp. 972–980.

[32] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *CVPR*, 2016, pp. 5695–5703.

[33] H. Park and K. M. Lee, "Look wider to match image patches with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1788–1792, 2017.

[34] R. Schuster, O. Wasenmuller, C. Unger, and D. Stricker, "SDC - stacked dilated convolution: A unified descriptor network for dense matching tasks," in *CVPR*, 2019.

[35] F. Zhang and B. W. Wah, "Fundamental principles on learning new features for effective dense matching," *TIP*, 2017.

[36] K. Batsos, C. Cai, and P. Mordohai, "CBMV: A coalesced bidirectional matching volume for disparity estimation," in *CVPR*, 2018, pp. 2060–2069.

[37] S. Tulyakov, A. Ivanov, and F. Fleuret, "Weakly supervised learning of deep metrics for stereo reconstruction," in *ICCV*, 2017.

[38] A. Spyropoulos, N. Komodakis, and P. Mordohai, "Learning to detect ground control points for improving the accuracy of stereo matching," in *CVPR*, 2014, pp. 1621–1628.

[39] A. Spyropoulos and P. Mordohai, "Correctness prediction, accuracy improvement and generalization of stereo matching using supervised learning," *IJCV*, vol. 118, no. 3, pp. 300–318, 2016.

[40] N. Komodakis, G. Tziritas, and N. Paragios, "Fast, approximately optimal solutions for single and dynamic MRFs," in *CVPR*, 2007.

[41] M.-G. Park and K.-J. Yoon, "Leveraging stereo matching with learning-based confidence measures," in *CVPR*, 2015, pp. 101–109.

[42] ——, "Learning and selecting confidence measures for robust stereo matching," *PAMI*, vol. 41, no. 6, pp. 1397–1411, 2018.

[43] M. Poggi and S. Mattoccia, "Learning a general-purpose confidence measure based on O(1) features and a smarter aggregation strategy for semi global matching," in *3DV*, 2016, pp. 509–518.

[44] M. Poggi, F. Tosi, and S. Mattoccia, "Learning a confidence measure in the disparity domain from O(1) features," *CVIU*, vol. 193, p. 102905, 2020.

[45] A. Seki and M. Pollefeys, "Patch based confidence prediction for dense disparity map." in *BMVC*, 2016, pp. 23.1–23.13.

[46] ——, "SGM-Nets: Semi-global matching with neural networks," in *CVPR*, 2017, pp. 231–240.

[47] J. L. Schönberger, S. N. Sinha, and M. Pollefeys, "Learning to fuse proposals from multiple scanline optimizations in semi-global matching," in *ECCV*, 2018, pp. 739–755.

[48] A. Shaked and L. Wolf, "Improved stereo matching with constant highway networks and reflective confidence learning," in *CVPR*, 2017, pp. 4641–4650.

[49] S. Gidaris and N. Komodakis, "Detect, replace, refine: Deep structured prediction for pixel wise labeling," in *CVPR*, 2017, pp. 5248–5257.

[50] X. Ye, J. Li, H. Wang, H. Huang, and X. Zhang, "Efficient stereo matching leveraging deep local and context information," *IEEE Access*, vol. 5, pp. 18 745–18 755, 2017.

[51] K. Batsos and P. Mordohai, "RecResNet: A recurrent residual CNN architecture for disparity map enhancement," in *3DV*, 2018, pp. 238–247.

[52] Z. Jie, P. Wang, Y. Ling, B. Zhao, Y. Wei, J. Feng, and W. Liu, "Left-right comparative recurrent model for stereo matching," in *CVPR*, 2018.

[53] P. Knöbelreiter and T. Pock, "Learned collaborative stereo refinement," in *German Conference on Pattern Recognition*, 2019, pp. 3–17.

[54] Z. Jie, P. Wang, Y. Ling, B. Zhao, Y. Wei, J. Feng, and W. Liu, "Left-right comparative recurrent model for stereo matching," in *CVPR*, 2018, pp. 3838–3846.

[55] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018.

[56] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *ICCV*, 2015, pp. 2758–2766.

[57] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *CVPR*, 2018.

[58] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[59] Z. Yin, T. Darrell, and F. Yu, "Hierarchical discrete distribution decomposition for match density estimation," in *CVPR*, 2019, pp. 6044–6053.

[60] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia, and L. Di Stefano, "Real-time self-adaptive deep stereo," in *CVPR*, June 2019.

[61] P. Knöbelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock, "End-to-end training of hybrid CNN-CRF models for stereo," in *CVPR*, 2017, pp. 2339–2348.

[62] 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 *ICCV*, 2017, pp. 887–895.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[64] Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou, and J. Zhang, "Learning for disparity estimation through feature constancy," in *CVPR*, 2018, pp. 2811–2820.

[65] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *NeurIPS*, 2018.

[66] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia, "SegStereo: Exploiting semantic information for disparity estimation," in *ECCV*, 2018, pp. 636–651.

[67] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[68] X. Song, X. Zhao, H. Hu, and L. Fang, "EdgeStereo: A context integrated residual pyramid network for stereo matching," in *ACCV*, 2018.

[69] X. Song, X. Zhao, L. Fang, H. Hu, and Y. Yu, "Edgestereo: An effective multi-task learning network for stereo matching and edge detection," *International Journal of Computer Vision*, pp. 1–21, 2020.

[70] W. Zhan, X. Ou, Y. Yang, and L. Chen, "DSNet: Joint learning for scene segmentation and disparity estimation," in *ICRA*, 2019, pp. 2946–2952.

[71] H. Jiang, D. Sun, V. Jampani, Z. Lv, E. Learned-Miller, and J. Kautz, "Sense: A shared encoder network for scene-flow estimation," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[72] C. Zhou, H. Zhang, X. Shen, and J. Jia, "Unsupervised learning of stereo matching," in *ICCV*, 2017, pp. 1567–1575.

[73] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *ICCV*, 2017, pp. 66–75.

[74] L. Yu, Y. Wang, Y. Wu, and Y. Jia, "Deep stereo matching with explicit cost aggregation sub-architecture," in *AAAI*, 2018.

[75] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *CVPR*, 2018, pp. 5410–5418.

[76] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *PAMI*, vol. 37, no. 9, pp. 1904–1916, 2015.

[77] G.-Y. Nie, M.-M. Cheng, Y. Liu, Z. Liang, D.-P. Fan, Y. Liu, and Y. Wang, "Multi-level context ultra-aggregation for stereo matching," in *CVPR*, 2019.

[78] X. Cheng, P. Wang, and R. Yang, "Learning depth with convolutional spatial propagation network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[79] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, "GA-Net: Guided aggregation net for end-to-end stereo matching," in *CVPR*, 2019.

[80] L. De-Maeztu, S. Mattoccia, A. Villanueva, and R. Cabeza, "Linear stereo matching," in *ICCV*, 2011, pp. 1708–1715.

[81] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *PAMI*, vol. 35, no. 2, pp. 504–511, 2012.

[82] R. Chabra, J. Straub, C. Sweeney, R. Newcombe, and H. Fuchs, "StereoDRNet: Dilated Residual StereoNet," in *CVPR*, 2019.

[83] C.-W. Xie, H.-Y. Zhou, and J. Wu, "Vortex pooling: Improving context representation in semantic segmentation," *arXiv preprint arXiv:1804.06242*, 2018.

[84] S. Tulyakov, A. Ivanov, and F. Fleuret, "Practical deep stereo (PDS): Toward applications-friendly deep stereo matching," in *NeurIPS*, 2018, pp. 5871–5881.

[85] 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 *ECCV*, 2018, pp. 573–590.

[86] Y. Wang, Z. Lai, G. Huang, B. H. Wang, L. van der Maaten, M. Campbell, and K. Q. Weinberger, "Anytime stereo image depth estimation on mobile devices," in *ICRA*, 2019.

[87] S. Liu, S. De Mello, J. Gu, G. Zhong, M.-H. Yang, and J. Kautz, "Learning affinity via spatial propagation networks," in *NeurIPS*, 2017, pp. 1520–1530.

[88] G. Yang, J. Manela, M. Happold, and D. Ramanan, "Hierarchical deep stereo matching on high-resolution images," in *CVPR*, 2019.

[89] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, "Group-wise correlation stereo network," in *CVPR*, 2019.

[90] Z. Wu, X. Wu, X. Zhang, S. Wang, and L. Ju, "Semantic stereo matching with pyramid cost volumes," in *ICCV*, 2019.

[91] S. Duggal, S. Wang, W.-C. Ma, R. Hu, and R. Urtasun, "Deeppruner: Learning efficient stereo matching via differentiable patchmatch," in *ICCV*, 2019.

[92] A. Li and Z. Yuan, "Occlusion aware stereo matching via cooperative unsupervised learning," in *ACCV*, 2018, pp. 197–213.

[93] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics*, vol. 28, no. 3, 2009.

[94] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017, pp. 270–279.

[95] F. Tosi, M. Poggi, A. Benincasa, and S. Mattoccia, "Beyond local reasoning for stereo confidence estimation with deep learning," in *ECCV*, 2018, pp. 319–334.

[96] M. Poggi, F. Tosi, and S. Mattoccia, "Quantitative evaluation of confidence measures in a machine learning world," in *ICCV*, 2017, pp. 5228–5237.

[97] R. Haeusler, R. Nair, and D. Kondermann, "Ensemble learning for confidence measures in stereo vision," in *CVPR*, 2013, pp. 305–312.

[98] X. Hu and P. Mordohai, "A quantitative evaluation of confidence measures for stereo vision," *PAMI*, vol. 34, no. 11, pp. 2121–2133, 2012.

[99] M. Poggi and S. Mattoccia, "Learning from scratch a confidence measure." in *BMVC*, 2016.

[100] ——, "Learning to predict stereo reliability enforcing local consistency of confidence maps," in *CVPR*, 2017, pp. 2452–2461.

[101] C. Mostegel, M. Rumpler, F. Fraundorfer, and H. Bischof, "Using self-contradiction to learn confidence measures in stereo vision," in *CVPR*, 2016.

[102] F. Tosi, M. Poggi, A. Tonioni, L. Di Stefano, and S. Mattoccia, "Learning confidence measures in the wild," in *BMVC*, 2017.

[103] Z. Fu and M. A. Fard, "Learning confidence measures by multimodal convolutional neural networks," in *WACV*, 2018, pp. 1321–1330.

[104] S. Kim, D. Min, S. Kim, and K. Sohn, "Feature augmentation for learning confidence measure in stereo matching," *TIP*, vol. 26, no. 12, pp. 6019–6033, 2017.

[105] R. Gouveia, A. Spyropoulos, and P. Mordohai, "Confidence estimation for superpixel-based stereo matching," in *3DV*, 2015, pp. 180–188.

[106] S. Kim, D. Min, S. Kim, and K. Sohn, "Unified confidence estimation networks for robust stereo matching," *TIP*, vol. 28, no. 3, pp. 1299–1313, 2018.

[107] S. Kim, S. Kim, D. Min, and K. Sohn, "LAF-Net: Locally adaptive fusion networks for stereo confidence estimation," in *CVPR*, 2019.

[108] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *NeurIPS*, 2015, pp. 2017–2025.

[109] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *NeurIPS*, 2017, pp. 5574–5584.

[110] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *CVPR*, 2018, pp. 7482–7491.

[111] J. Gast and S. Roth, "Lightweight probabilistic deep networks," in *CVPR*, 2018, pp. 3369–3378.

[112] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "On the uncertainty of self-supervised monocular depth estimation," in *CVPR*, 2020.

[113] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia, "Learning monocular depth estimation infusing traditional stereo knowledge," in *CVPR*, 2019.

[114] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.

[115] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *ICML*, 2018, pp. 1989–1998.

[116] A. Tonioni, M. Poggi, S. Mattoccia, and L. Di Stefano, "Unsupervised adaptation for deep stereo," in *ICCV*, 2017, pp. 1605–1613.

[117] ——, "Unsupervised domain adaptation for depth prediction from images," *PAMI*, 2019.

[118] P. Knöbelreiter, C. Vogel, and T. Pock, "Self-supervised learning for stereo reconstruction on aerial images," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 4379–4382.

[119] J. Pang, W. Sun, C. Yang, J. Ren, R. Xiao, J. Zeng, and L. Lin, "Zoom and learn: Generalizing deep stereo matching to novel domains," in *CVPR*, 2018, pp. 2070–2079.

[120] Y. Zhong, H. Li, and Y. Dai, "Open-world stereo video matching with deep RNN," in *ECCV*, 2018, pp. 101–116.

[121] M. Poggi, D. Pallotti, F. Tosi, and S. Mattoccia, "Guided stereo matching," in *CVPR*, 2019.

[122] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *PAMI*, vol. 31, no. 5, pp. 824–840, 2009.
[123] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *CVPR*, 2014, pp. 89–96.
[124] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *PAMI*, vol. 38, no. 10, pp. 2024–2039, 2016.
[125] K. Karsch, C. Liu, and S. B. Kang, "Depth transfer: Depth extraction from video using non-parametric sampling." *PAMI*, vol. 36, no. 11, pp. 2144–2158, 2014.
[126] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *3DV*, 2016, pp. 239–248.
[127] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.
[128] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *TIP*, vol. 13, no. 4, pp. 600–612, 2004.
[129] J. P. Yusiong and P. Naval, "AsiANet: Autoencoders in Autoencoder for Unsupervised Monocular Depth Estimation," in *WACV*, 2019, pp. 443–451.
[130] M. Poggi, F. Tosi, and S. Mattoccia, "Learning monocular depth estimation with unsupervised trinocular assumptions," in *3DV*, 2018.
[131] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *NeurIPS*, 2014, pp. 2366–2374.
[132] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, 2017.
[133] C. Godard, O. Mac Aodha, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *ICCV*, 2019.
[134] F. Aleotti, F. Tosi, M. Poggi, and S. Mattoccia, "Generative adversarial networks for unsupervised monocular depth prediction," in *ECCV Workshops*, 2018.
[135] M. M. Puscas, D. Xu, A. Pilzer, and N. Sebe, "Structured coupled generative adversarial networks for unsupervised monocular depth estimation," in *3DV*, 2019.
[136] I. Mehta, P. Sakurikar, and P. J. Narayanan, "Structured adversarial training for unsupervised monocular depth estimation," in *3DV*, 2018.
[137] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "Towards real-time unsupervised monocular depth estimation on cpu," in *IEEE/JRS Conference on Intelligent Robots and Systems (IROS)*, 2018.
[138] S. Pillai, R. Ambrus, and A. Gaidon, "Superdepth: Self-supervised, super-resolved monocular depth estimation," in *ICRA*, 2019.
[139] A. Pilzer, S. Lathuiliere, N. Sebe, and E. Ricci, "Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation." in *CVPR*, 2019.
[140] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *CVPR*, 2018.
[141] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille, "Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding," *PAMI*, 2019.
[142] N. Yang, R. Wang, J. Stückler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in *ECCV*, 2018, pp. 835–852.
[143] J. Watson, M. Firman, G. J. Brostow, and D. Turmukhambetov, "Self-supervised monocular depth hints," in *ICCV*, 2019.
[144] Y. Kuznietsov, J. Stuckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *CVPR*, 2017.
[145] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin, "Single view stereo matching," in *CVPR*, 2018, pp. 155–163.
[146] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang, "Learning monocular depth by distilling cross-domain stereo networks," in *ECCV*, 2018, pp. 484–500.
[147] P. Z. Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano, "Geometry meets semantics for semi-supervised monocular depth estimation," in *ACCV*, 2018, pp. 298–313.
[148] L. Andraghetti, P. Myriokefalitakis, P. L. Dovesi, B. Luque, M. Poggi, A. Pieropan, and S. Mattoccia, "Enhancing self-supervised monocular depth estimation with traditional visual odometry," in *3DV*, 2019.
[149] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.
[150] J. Xie, R. Girshick, and A. Farhadi, "Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks," in *ECCV*, 2016, pp. 842–857.
[151] V. Peluso, A. Cipolletta, A. Calimera, M. Poggi, F. Tosi, and S. Mattoccia, "Enabling energy-efficient unsupervised monocular depth estimation on armv7-based platforms," in *Design Automation and Test in Europe (DATE)*, 2019.
[152] P. Zama Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano, "Geometry meets semantic for semi-supervised monocular depth estimation," in *ACCV*, 2018.
[153] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, "Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding," in *ECCV Workshops*, 2018.

Matteo Poggi received his PhD degree in Computer Science and Engineering from University of Bologna 2018. Currently, he is a Post-doc researcher at Department of Computer Science and Engineering, University of Bologna. His research interests include deep learning for depth estimation and embedded computer vision. He is the author of more than 30 papers about these topics.

Fabio Tosi earned his MS degree from the University of Bologna in 2017. Currently he is a PhD student in Computer Science and Engineering at the University of Bologna, working on deep learning for stereo and monocular depth estimation.

Konstantinos Batsos earned his MS and PhD from Stevens Institute of Technology in 2011 and 2020, respectively. His research interests include binocular and multi-view stereo, deep learning and real-time computer vision. He is currently a software engineer at Argo AI.

Philippos Mordohai is a professor of Computer Science at Stevens Institute of Technology. He earned his PhD from the University of Southern California and held postdoctoral appointments at the University of North Carolina and the University of Pennsylvania. His research interests span 3D reconstruction from images and video, range data analysis, perceptual organization and active vision. He serves as an associate editor for the IEEE Transactions on Pattern Analysis and Machine Intelligence, Computer Vision and Image Understanding, and Image and Vision computing. He was a program co-chair of the International Conference on 3D Vision (3DV), 2019.

Stefano Mattoccia received a Ph.D. degree in Computer Science Engineering from the University of Bologna in 2002. Currently he is an associate professor at the Department of Computer Science and Engineering of the University of Bologna. His research interest is mainly focused on computer vision, depth perception from images, deep learning and embedded computer vision.