Learning End-to-End Scene Flow by Distilling Single Tasks Knowledge

Filippo Aleotti, Matteo Poggi, Fabio Tosi, Stefano Mattoccia
Department of Computer Science and Engineering (DISI)
University of Bologna, Italy

Abstract

Scene flow is a challenging task aimed at jointly estimating the 3D structure and motion of the sensed environment. Although deep learning solutions achieve outstanding performance in terms of accuracy, these approaches divide the whole problem into standalone tasks (stereo and optical flow) addressing them with independent networks. Such a strategy dramatically increases the complexity of the training procedure and requires power-hungry GPUs to infer scene flow barely at 1 FPS. Conversely, we propose DWARF, a novel and lightweight architecture able to infer full scene flow jointly reasoning about depth and optical flow easily and elegantly trainable end-to-end from scratch. Moreover, since ground truth images for full scene flow are scarce, we propose to leverage on the knowledge learned by networks specialized in stereo or flow, for which much more data are available, to distill proxy annotations. Exhaustive experiments show that i) DWARF runs at about 10 FPS on a single high-end GPU and about 1 FPS on NVIDIA Jetson TX2 embedded at KITTI resolution, with moderate drop in accuracy compared to 10× deeper models, ii) learning from many distilled samples is more effective than from the few, annotated ones available.

1 Introduction

The term *Scene Flow* refers to the three-dimensional dense motion field of a scene (Vedula et al. 1999) and enables to effectively model both 3D structure and movements of the sensed environment, crucial for a plethora of high-level tasks such as augmented reality, 3D mapping and autonomous driving. Dense scene flow inference requires the estimation of two crucial cues for each observed point: depth and motion across frames acquired over time. Such cues can be obtained deploying two well-known techniques in computer vision: stereo matching and optical flow estimation. The first one aims at inferring the disparity (i.e. depth) by matching pixels across two rectified images acquired by synchronized cameras, the second at determining the 2D motion between corresponding pixels across two consecutive frames, thus requiring at least four images for full scene flow estimation. For years, solutions to scene flow (Behl et al. 2017) have been rather accurate, yet demanding in terms of computational costs and runtime. Meanwhile, deep learning established as state-of-the-art for stereo matching (Mayer et al. 2016) and optical flow (Dosovitskiy et al. 2015). Thus, more recent approaches to scene flow leveraged this novel paradigm stacking together single-task architectures (Ilg et al. 2018; Ma et al. 2019). However, this strategy is demanding as well and requires separate and specific training for each network and does not fully exploit the inherent dependency between the tasks, e.g. the flow of 3D objects depends on their distance, their motion and camera ego-motion (Taniai, Sinha, and Sato 2017), as a single model could. On the other hand, either synthetic or real datasets annotated with full scene flow labels are rare compared to those disposable for stereo and flow alone. This constraint limits the *knowledge* available to a single network compared to the one exploitable by an ensemble of specialized ones.

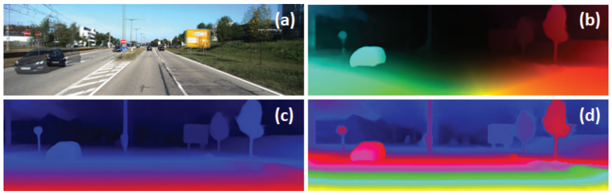

Figure 1: End-to-end scene flow with DWARF. (a) Superimposed reference images at time t_1 and t_2, (b) estimated optical flow, (c) disparity and (d) disparity change.

To tackle previous issues, in this paper, we propose a novel lightweight architecture for scene flow estimation jointly inferring disparity, optical flow and disparity change (i.e., the depth component of 3D motion). We design a custom layer, namely 3D correlation layer, by extending the formulation used to tackle the two tasks singularly (Dosovitskiy et al. 2015; Mayer et al. 2016), in order to encode matching relationships across the four images. Moreover, to overcome the constraint on training data, we recover the missing knowledge leveraging standalone, state-of-the-art networks for stereo and flow to generate proxy annotations for our single scene flow architecture. Using this strategy on the KITTI dataset (Menze and Geiger 2015), we *distill* about 20× samples compared to the number of ground truth images available, enabling for more effective training and thus to more accurate estimations.

Our architecture for scene flow estimation through Disparity, Warping and Flow (dubbed **DWARF**) can be elegantly trained in an end-to-end manner from scratch and yields competitive results compared to state-of-the-art, although running about 10× faster thanks to efficient design strategies. Figure 1 shows a qualitative example of dense scene flow estimation achieved by our network, enabling 10 FPS on NVIDIA Titan 1080Ti and about 1 FPS on Jetson TX2 embedded system.

2 Related Work

We review the literature concerning deep learning for optical flow and stereo, as well as scene flow estimation.

Optical flow. Starting from the seminal work by Horn and Schunck (1981), many others researchers mainly tackled optical flow deploying variational (Sun, Roth, and Black 2014; Revaud et al. 2015) and learning based (Sun et al. 2008; Wulff and Black 2015) approaches. Nonetheless, starting from FlowNet (Dosovitskiy et al. 2015) most recent works rely on deep learning. Specifically, it introduces the design of a 2D correlation layer encoding similarity between pixels, rapidly becoming a standard component in end-to-end networks for flow and stereo. The results obtained by FlowNet have been improved stacking more networks (Ilg et al. 2017), significantly increasing the number of parameters of the overall model. SpyNet (Ranjan and Black 2017) addresses the complexity issue through coarse-to-fine optical flow estimation. PWCNet (Sun et al. 2017) and LiteFlowNet (Hui, Tang, and Loy 2018) further improved this strategy using a correlation layer at each stage of the pyramid. Finally, self-supervised optical flow has been studied by leveraging view synthesis (Meister, Hur, and Roth 2018) or by distilling labels in a teacher-student scheme (Liu et al. 2019).

Stereo matching. Inferring depth from stereo pairs is a long-standing problem in computer vision, and well-known geometric constraints can be exploited to estimate disparity and then to obtain depth by triangulation. Although traditional methods such as SGM (Hirschmuller 2005) are a popular choice, deep learning gave a notable boost in accuracy and it represents the state-of-the-art. Zbontar and LeCun (2016) replaced conventional matching costs computation with a siamese CNN network. Luo, Schwing, and Urtasun (2016) cast the correspondence problem as a multi-class classification task. Mayer et al. (2016) introduced *DispNetC*, an end-to-end trainable network leveraging on the same correlation layer proposed for flow (Dosovitskiy et al. 2015), but applied to 1D domain. Kendall et al. (2017) proposed to stack a cost volume and to process it with 3D convolutions. Following these latter two design strategies, many works have been proposed leveraging correlation scores (Liang et al. 2018; Yang et al. 2018; Song et al. 2018), 3D convolutions (Chang and Chen 2018; Zhang et al. 2019) or both (Guo et al. 2019) to further improve the final accuracy. Poggi et al. (2019) improved end-to-end stereo with LiDAR guidance. As for optical flow, coarse-to-fine strategies and warping (Tonioni et al. 2019b; Yin, Darrell, and Yu 2019; Dovesi et al. 2019) yielded compact, yet accurate, architectures suited even for embedded devices.

Scene flow. Vedula et al. (1999) represents the very first attempt to estimate scene flow from multi-view frame sequences. Most times this task has been cast as a variational problem (Wedel et al. 2011; Pons, Keriven, and Faugeras 2007; Basha, Moses, and Kiryati 2010; Zhang and Kambhamettu 2001; Huguet and Devernay 2007; Valgaerts et al. 2010). Some works applied 3D regularisation techniques based on a model of dense depth and 3D motion (Vogel, Schindler, and Roth 2011) and a piecewise rigid scene model (better known as PRSM) for stereo (Vogel, Schindler, and Roth 2013) and multi-frame (Vogel, Schindler, and Roth 2015). (Behl et al. 2017) involve object and instance recognition performed by a CNN into scene flow estimation within a CRF-based model, Menze and Geiger (2015) tackled scene flow estimation by segmenting the scene into multiple rigidly-moving objects. Taniai, Sinha, and Sato (2017) deployed a pipeline for multi-frame scene flow estimation, visual odometry and motion segmentation running in a couple of seconds. Concerning deep learning methods, Ilg et al. (2018) trained specialised networks for stereo and optical flow and combined them within a refinement module to estimate disparity change. Similarly, Ma et al. (2019) rely on pre-existing networks for stereo (Chang and Chen 2018), optical flow (Sun et al. 2017) and segmentation (He et al. 2017) and then infer scene flow through a Gaussian-Newton solver. Despite much faster compared to the prior state-of-the-art, they require multi-stage training protocols for every single task and power-hungry GPUs to barely run at 1 FPS.

Recently, Saxena et al. (2019) proposed a fast and lightweight model taking into account occlusions. Although similar in design, we will show that DWARF outperforms it by a good margin. Concurrent to our work, Jiang et al. (2019) propose a similar architecture to jointly learn for scene flow and semantics.

3 Proposed Architecture

In this section, we introduce the DWARF architecture built upon established principles from optical flow and stereo matching to obtain, in synergy, an end-to-end framework for full scene flow estimation. As already proved in different fields, coarse-to-fine designs enable for compact, yet accurate models.

Given a couple of stereo image pairs L_1, R_1, L_2 and R_2 referencing, respectively, the left and right images at time t_1 and t_2 we aim at estimating disparity $\mathcal{D}_1$ between L_1, R_1 to obtain its 3D position at time t_1, optical flow $\mathcal{F}_1$ between L_1, L_2 to get 2D motion vectors connecting pixels in L_1 to those in L_2 and disparity change $\mathcal{D}_{1\leftarrow 2}$, i.e. disparity $\mathcal{D}_2$ between L_2, R_2 mapped on corresponding pixels in image L_1 that allows to get z component of 3D motion vectors. To achieve this, our model performs a first extraction phase in order to retrieve a pyramid of features from each image, then in a coarse-to-fine manner it computes point-wise correlations across the four features representations and estimates the aforementioned disparity and motion vectors, going up to the last level of the pyramid to obtain the final output. Figure 2 sketches the structure of DWARF configured to process, for the sake of space, a pyramid down to $\frac{1}{32}$ of the original resolution. In the next section, we will describe in detail each module depicted in the figure.

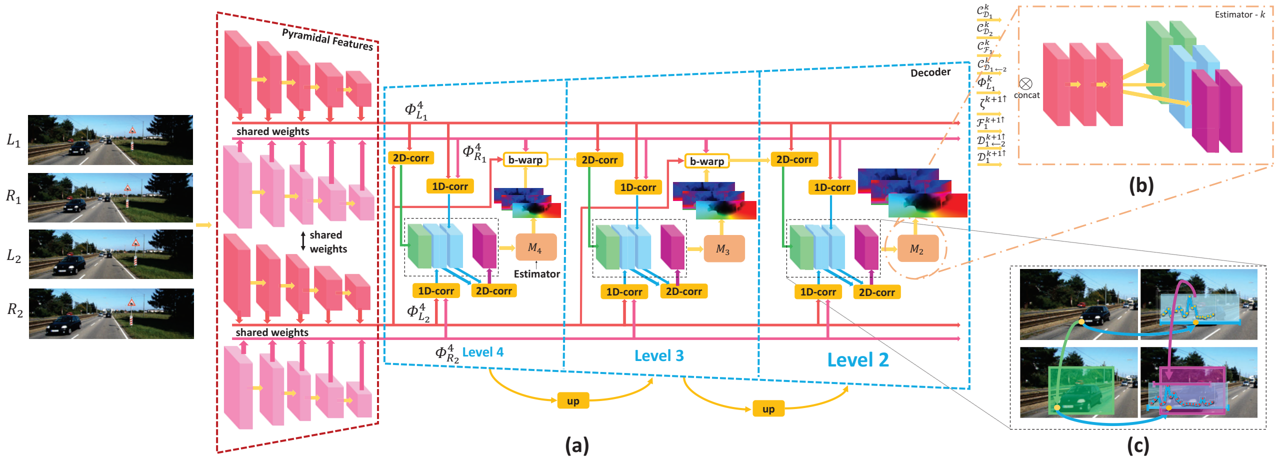

Figure 2: DWARF architecture. The full architecture (a) has shared encoders (pink) to extract pyramids of features. At each resolution k, correlation scores respectively in green, light blue, light blue and purple, are stacked and forwarded to the estimator to generate $\mathcal{F}^k_1$, $\mathcal{D}^k_1$ and $\mathcal{D}^k_{1\leftarrow 2}$. Such outputs are used to warp features at level $k-1$ until the final resolution is reached. Each estimator (b) is made of a common backbone followed by three task-specific heads. Correlation layers encode matching between pixels across the four images (c).

3.1 Features Extraction

To extract meaningful representations from each input image, we design a compact encoder to obtain a pyramid of features ready to be processed in a coarse-to-fine manner. Purposely, DWARF has four encoders, one for each input image, with shared weights. Each one is built of a block of three 3×3 convolutional layers for each level in the pyramids of features, respectively with stride 2, 1 and 1. For the sake of space, Figure 2 (a) shows an example of 5 levels encoder. Actually DWARF deploys a 6 levels encoder down to $\frac{1}{64}$ resolution features (k=6), counting 18 convolutional layers, each followed by Leaky ReLU activations with $\alpha = 0.1$. By progressively decimating the spatial dimensions, we increase the amount of extracted features, respectively to 16, 32, 64, 96, 128 and 196. It generates features $\phi^k_{L_1}, \phi^k_{R_1}, \phi^k_{L_2}$ and $\phi^k_{R_2}$ with $k \in [1, 6]$, respectively for frames L_1, R_1, L_2 and R_2, deployed by the following module to extract matching relationships between pixels.

3.2 Warping

The main advantage introduced by a coarse-to-fine strategy consists of computing small disparity and flow vectors at each resolution and sum them while going up the pyramid. This strategy allows keeping a small range where to calculate correlation scores, as we will discuss in detail in the next section. Otherwise, a large search space would dramatically increase the complexity of the entire network.

Given features $\phi^k_{L_1}, \phi^k_{R_1}, \phi^k_{L_2}$ and $\phi^k_{R_2}$ extracted by the encoder at the k^{th} level, we have to bring all features closer to $\phi^k_{L_1}$ coordinates. To do so, estimates at previous pyramid level $(k+1)$ are upsampled, e.g. $\mathcal{D}^{k+1}_1$ to $\mathcal{D}^{k+1^\uparrow}_1$, and properly scaled to match stereo/flow at the next resolution k. Then, features are warped by means of backward warping, in particular $\phi^k_{L_2}$ according to optical flow $\mathcal{F}^{k+1^\uparrow}_1$ and $\phi^k_{R_1}$ according to $\mathcal{D}^{k+1^\uparrow}_1$. Finally, the motion that allows to warp $\phi^k_{R_2}$ towards $\phi^k_{L_1}$ is given by the sum of $\mathcal{F}^{k+1^\uparrow}_{L_1}$ and $\mathcal{D}^{k+1^\uparrow}_{1\leftarrow 2}$. This because the former encodes the mapping between present and future correspondences, while the latter the horizontal displacement occurring between L_2 and R_2, but on L_1 coordinate, thus the same as $\mathcal{F}^{k+1^\uparrow}_1$.

We will see how this translates into computing, at each resolution k, a refined scene flow field to ameliorate a prior, coarse estimation inferred at resolution $k+1$. At the lowest resolution in the pyramid, features are not warped since scene flow priors are not available.

3.3 Cost Volumes and 3D Correlation Layer

Since DWARF jointly reasons about stereo and optical flow, correlation layers fit very well in its design. At first we compute correlation scores encoding standalone tasks, i.e. estimation of disparity $\mathcal{D}_1$ between L_1, R_1, $\mathcal{D}_2$ between L_2, R_2 and flow $\mathcal{F}_1$ between L_1, L_2, obtaining $\mathcal{C}^k_{\mathcal{D}_1}(\phi^k_{L_1}, \phi^k_{R_1})$, $\mathcal{C}^k_{\mathcal{D}_2}(\phi^k_{L_2}, \phi^k_{R_2})$, $\mathcal{C}^k_{\mathcal{F}_1}(\phi^k_{L_1}, \phi^k_{L_2})$ by means of two 1D and one 2D correlation layers depicted in light blue and green in Figure 2 (a). By defining the correlation between per-pixel features as $\langle\cdot\rangle$ and concatenation as $\otimes$, we obtain 1D and 2D correlations as

$$
\begin{aligned}
\mathcal{C}^k_{\mathcal{D}_t}(y, x) &= \bigotimes_{j \in [-r_x, r_x]} \langle \phi^k_{L_t}(y, x), \phi^k_{R_t}(y, x+j)_w \rangle \\
\mathcal{C}^k_{\mathcal{F}_1}(y, x) &= \bigotimes_{\substack{i \in [-r_y, r_y], \\ j \in [-r_x, r_x]}} \langle \phi^k_{L_1}(y, x), \phi^k_{L_2}(y+i, x+j)_w \rangle
\end{aligned} \tag{1}
$$

with (y, x) pixel coordinates, r_y, r_x radius on y and x directions, $t \in [1, 2]$. Subscript w means warping via upsampled priors as described in Section 3.2. Although such features embody relationships about standalone tasks, they lack at encoding matching between the 3D motion of the scene. To overcome this limitation, we introduce a novel custom layer.

Figure 2 (c) depicts how correlation layers act in DWARF. While 2D correlation layer (green) encodes similarities between pixels aimed at estimating optical flow, 1D correlations (light blue) compute scores between left and right images independently from time. Each produces a correlation curve, superimposed on R_1 and R_2 in the figure. If a pixel does not change its disparity through time, the peaks in the two curves would ideally match. Otherwise, they will appear shifted by the magnitude of the disparity change. The rest of the curve will shrink/enlarge, with major differences in portions dealing with regions moving of different motions (e.g., background vs foreground objects). This pattern, if properly learned, acts as a bridge between depth and 2D flow, enabling to infer the full 3D motion. Unfortunately, this behaviour is not explicitly modelled by the layers mentioned above. Hence, we adopt a novel component, namely a 3D correlation layer, whose search volume is depicted in purple in Figure 2 (c). Since correlation curves are already available from 1D correlation layers, this translates into computing *correlations over correlations* volumes as

$$\mathcal{C}^k_{\mathcal{D}_{1\leftarrow 2}} = \bigotimes_{\substack{i\in[-r_y,r_y],\\ j\in[-r_x,r_x],\\ h\in[-r_z,r_z]}} \langle \mathcal{C}^k_{\mathcal{D}_1}(y,x,d), \mathcal{C}^k_{\mathcal{D}_2}(y+i,x+j,d+h)\rangle \tag{2}$$

with (y, x, d) pixel coordinates in the correlation volumes and r_z the search radius for displacement between 1D correlation curves. Specifically, the full search space of such operation is 3D, being it over pixel coordinates plus displacement between correlation curves. We refer the reader to the **supplementary material** for some examples supporting this rationale, while ablation studies reported among our experiments prove the effectiveness of such a new layer, allowing DWARF to outperform similar architectures (Saxena et al. 2019) on the KITTI online benchmark.

3.4 Scene Flow Estimation

After the extraction of meaningful correlation features, we stack them into a features volume forwarded to a compact decoder network in charge of estimating the three components of the scene flow. As shown in Figure 2 (b), at each level the volume contains reference image features $\phi^k_{L_0}$, correlation scores, upsampled scene flow priors and latest features ζ extracted before estimation at level $k+1$. This input is forwarded to level k decoder. First, three convolutional layers with respectively 128, 128 and 96 channels rearrange the volume. Then, three independent *heads* are in charge of predicting $\mathcal{D}^k_1$, $\mathcal{F}^k_1$ and $\mathcal{D}^k_{1\leftarrow 2}$. Following this design, the network is forced to create a first holistic representation of the volume, then specialized by each sub-module. Each head has two task-specific 3×3 convolutional layers with 64 and 32 channels, producing ζ features from which a final 3×3 layer extracts the single component of scene flow at level k, e.g. $\mathcal{D}^k_1$. Such estimates, together with features ζ, are upsampled through a transposed convolution layer with stride 2, to provide coarse scene flow priors for warping at level $k-1$.

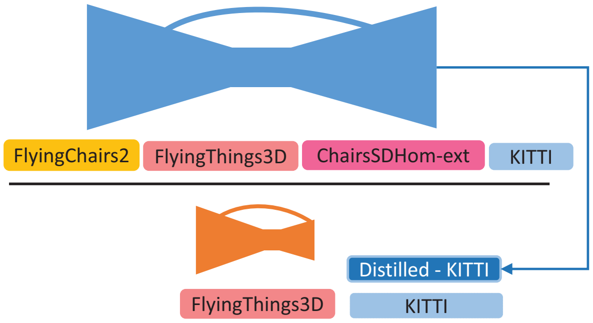

Figure 3: Knowledge distillation (Hinton, Vinyals, and Dean 2015) scheme. From an ensemble of deep networks (Ilg et al. 2018) (blue) trained on a variety of datasets we transfer knowledge to our compact model (orange).

Leaky ReLU units with $\alpha = 0.1$ follow all layers. Each estimator is optionally designed with dense connections (Huang et al. 2017) to boost accuracy. This design choice adds about 10 million parameters to DWARF.

3.5 Residual Refinement

Although the explicit reasoning about features matching across the four images is an effective way to guide the network towards scene flow estimation, it has limitations for pixels having missing correspondences. This fact occurs when, in one or multiple frames, they are occluded or no longer part of the observed scene. For instance, portions of the sensed scene located near image borders at time t_1 are no longer framed at t_2 when the camera is moving. To soften this problem, three residual networks are deployed to refine each single component of the full scene flow estimates, taking as input ζ features from the top-level estimator and processing it with six 3×3 convolutional layers extracting respectively 128, 128, 128, 96, 64, 32 features, with a dilation factor of 1, 2, 4, 8, 16, and 1 respectively to increase the receptive field introducing moderate overhead. A Leaky ReLU with $\alpha = 0.1$ follows each layer. Then, a further 3×3 convolutional layer (without activation units) extracts residual scene flow, summed to previous final estimations in order to refine them.

3.6 Knowledge Distillation from Expert Models

As previously pointed out, although end-to-end training is elegant and easier to schedule, it prevents using task-specific datasets since ground truth labels are required for full scene flow. However, a proper training schedule across several datasets is needed to achieve the best accuracy on single tasks (Ilg et al. 2017; Sun et al. 2017). To overcome this limitation, we leverage on knowledge distillation (Hinton, Vinyals, and Dean 2015) employing expert models trained for the single tasks and used to teach to a student network, DWARF in this case.

Specifically, we choose the ensemble of networks proposed by Ilg et al. (2018) to guide our simpler model, thanks to the availability of the source code and its excellent performance. Firstly a FlowNet-CSS and DispNet-CSS are in charge of optical flow and disparity estimation, then a third FlowNet-S architecture processes disparity $\mathcal{D}_2$ back warped according to computed optical flow and refines it to obtain $\mathcal{D}_{1\leftarrow 2}$. The three networks are trained in a multi-stage manner, starting from DispNet-CSS and FlowNet-CSS, and ending with the training of the final FlowNet-S. This allows for multi-dataset training, especially in the case of optical flow for which several sequential rounds of training on FlyingChairs2 and ChairsSDHom-ext (Ilg et al. 2017) are performed to achieve the best accuracy. By teaching DWARF with the expert models, we are able to both i) bring the knowledge learned by the expert model on task-specific datasets (e.g., FlyingChairs2 and ChairsSDHom-ext) to our model and ii) distill an extended training set, counting a larger amount and more variegated samples. We will show in our experiments how the knowledge distillation scheme results more effective than training on the few ground truth images available from real datasets.

3.7 Training Loss

Given the set of learnable parameters of the network Θ, $\mathcal{D}_1^k(\Theta)$, $\mathcal{D}_{1\leftarrow 2}^k(\Theta)$ and $\mathcal{F}_1^k(\Theta)$ respectively the estimated disparity, disparity change and optical flow, $\mathcal{D}_1^k(GT)$, $\mathcal{D}_{1\leftarrow 2}^k(GT)$ and $\mathcal{F}_1^k(GT)$ the ground truth maps for specific scene flow components brought to each pyramid level k, we adopt the L1 norm to optimise DWARF:

$$\mathcal{L}(\Theta) = \gamma\|\Theta\|_1^2 + \epsilon_1 \sum_{k=l_1}^{L} \alpha_k \|\mathcal{D}_1(\Theta) - \mathcal{D}_1^k(GT)\|_1$$

$$+\epsilon_2 \sum_{k=l_1}^{L} \alpha_k \|\mathcal{D}_{1\leftarrow 2}^k(\Theta) - \mathcal{D}_{1\leftarrow 2}^k(GT)\|_1 + \epsilon_3 \sum_{k=l_1}^{L} \alpha_k \|\mathcal{F}_1^k(\Theta) - \mathcal{F}_1^k(GT)\|_1 \tag{3}$$

We deploy a 6 levels pyramidal structure, extracting features up to level 6, halving the spatial resolution down to $\frac{1}{64}$. We set $l_0 = 2$, thus estimating scene flow up to quarter resolution and then bilinearly upsampling to the original input resolution. This strategy allows us to keep low memory requirements and fast inference time. The search spaces are set to 9, 9×9 and $9 \times 9 \times 1$ respectively for 1D, 2D and 3D correlations. A search range of 1 on disparity change keeps low the overall complexity of the network, yet significantly improving the accuracy on all metrics.

4 Experimental Results

We report extensive experiments aimed at assessing the accuracy and performance of DWARF. First, we describe in detail the training schedules. Then, we conduct an ablation study to measure the contribution of each component and compare DWARF to state-of-the-art deep learning approaches. Finally, we focus on DWARF run-time performance, extensively studying its behaviour on a variety of hardware platform, including a popular embedded device equipped with a low-power GPU.

4.1 Training Datasets and Protocol

It is a common practice to initialize end-to-end networks on large synthetic datasets before fine-tuning on real data (Dosovitskiy et al. 2015; Mayer et al. 2016; Ilg et al. 2017). Despite the large availability of synthetic datasets for flow and stereo (Butler et al. 2012; Dosovitskiy et al. 2015; Mayer et al. 2016), only the one proposed in (Mayer et al. 2016) provides ground truth for full scene flow estimation. In this field, KITTI 2015 (Menze and Geiger 2015) represents the unique example of a realistic benchmark for scene flow. Therefore, we scheduled training on these two datasets and, optionally, we leverage knowledge distillation (Hinton, Vinyals, and Dean 2015) from an expert network (Ilg et al. 2018) to augment the variety of realistic samples and consequently to better train DWARF.

Flying Things 3D. We set $\alpha_6 = 0.32$, $\alpha_5 = 0.08$, $\alpha_4 = 0.02$, $\alpha_3 = 0.01$ and $\alpha_2 = 0.005$, $\gamma = 0.0004$ and cross-task weights to $\epsilon_1 = 1$, $\epsilon_2 = 1$ and $\epsilon_3 = 0.5$. Ground truth values are down-scaled to match the resolution of the level and scaled by a factor of 20, as done by Dosovitskiy et al.; Sun et al. (2015; 2017). The network has been trained for 1.2M steps with a batch size of 4 randomly selecting crops with size 768×384, using Adam optimiser (Kingma and Ba 2014), with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and initial learning rate of 10^{-4}, which has been halved after 400K, 600K, 800K and 1M steps.

KITTI 2015. We fine-tuned the network using the 200 training images from the KITTI Scene Flow (Menze, Heipke, and Geiger 2015) dataset with a batch size of 4 for 50K steps. Again, Adam optimizer (Kingma and Ba 2014) has been adopted with the same parameters as before. The initial learning rate is set to 3×10^{-5}, halved after 25K, 35K and 45K steps. We minimise loss only at level $k = 2$. Specifically, we upsample through bilinear interpolation the predictions at the quarter resolution and apply the fine-tuning loss described in 3.7 at full resolution. Predictions at lower levels have not been optimized explicitly. We set $\epsilon_1 = 1$, $\epsilon_2 = 1$ and $\epsilon_3 = 0.5$, all the α_k set to 0 with the exception of α_2 set to 0.001 while γ is left untouched. Images are firstly padded to 1280×384 pixels, then random crops of size 896×320 are extracted at each iteration.

Distilled-KITTI. Finally, we perform knowledge distillation to produce an extended set of images for fine-tuning DWARF. Specifically, we use the 4000 total images available from the multiview extension of the KITTI 2015 training set and we produce proxy annotations leveraging FlowNet-CSS, DispNet-CSS and FlowNet-S. We use the trained models made available by the authors, trained on multiple task-specific datasets and fine-tuned on the aforementioned KITTI 2015 split (i.e., 200 images). We point out that, excluding the task-specific synthetic datasets, the expert models are trained with the same real ground truth (i.e., no additional annotations) and are used only to distill more *proxy* labels. Moreover, despite the extremely accurate estimates produced by the expert models on the KITTI training split (below 2% error rate on full scene flow), the labels sourced through distillation are yet *noisy*.

Data augmentation. We perform data augmentation by applying random gamma correction in [0.8,1.2], additive brightness in [0.5,2.0], and colour shifts in [0.8,1.2] for each channel separately. To increase robustness against brightness changes, we applied augmentation independently to every

Configuration			Params	Flow	Disparity	Change
Dense	3Dcorr	Refine	M	EPE	EPE	EPE
			5.06	7.435	1.959	2.283
✓			13.50	6.758	1.837	2.092
✓	✓		15.87	6.738	1.827	2.149
✓	✓	✓	19.62	**6.440**	**1.784**	**2.039**

Table 1: Ablation study on the FlyingThings3D test set. For each variant of DWARF, we report End Point Error (EPE) for flow, disparity and change respectively.

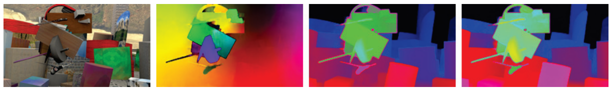

Figure 4: Qualitative results on FlyingThings 3D (Mayer et al. 2016) test split. From left to right, reference image at t_1, $\mathcal{F}_1$, $\mathcal{D}_1$ and $\mathcal{D}_{1\leftarrow 2}$.

Configuration							
Dense	3Dcorr	Refine	Sup.	F1-All	D1-All	D2-All	SF-All
✓	✓	✓	Gt	18.53	4.58	9.32	20.85
✓	✓	✓	Px	20.71	3.94	9.14	23.07
✓	✓	✓	Px + Gt	20.47	**3.91**	9.43	23.01
✓	✓	✓	Px → Gt	**16.75**	4.22	**8.26**	**19.00**

Table 2: Impact of knowledge distillation and its scheduling on 40 images from KITTI training set. We report the percentage of pixels with error higher than 3 and 5% respectively for flow, disparity, change and full scene flow.

Configuration			Jetson TX2			1080 Ti
Dense	3DCorr	Refine	Max-Q	Max-P	Max-N	(≈250W)
			0.79s	0.65s	0.57s	0.09s
✓			1.26s	1.05s	0.91s	0.10s
✓	✓		1.47s	1.22s	1.06s	0.11s
✓	✓	✓	2.21s	1.83s	1.59s	0.14s

Table 3: Runtime analysis for different variants of DWARF on NVIDIA Jetson TX2 (using Max-Q, Max-P, Max-N configurations) and NVIDIA GTX 1080Ti. Time in seconds.

single image. Instead, random zooming, with probability 0.5 of re-scaling the image by a random factor in [1,1.8], has been applied in the same way to L_1, R_1, L_2 and R_2 and the relative ground truths.

4.2 Ablation Studies

In this section, we study the effectiveness of each architectural choice. Tables 1 and 2 report experimental results on FlyingThings3D (Mayer et al. 2016) test set and KITTI 2015 training set (Menze, Heipke, and Geiger 2015) by i) increasing the complexity of the network and ii) introducing the knowledge distillation process.

FlyingThings3D. This dataset provides 4248 frames for validation. In Table 1 we report average End-Point-Error (EPE) for the disparity, flow and change (respectively D1, F1 and D2) on 3822 images, obtained by filtering the validation set according to the guidelines. We trained four DWARF variants, starting from the simple *Standalone tasks* version (i.e., without Dense, 3D correlation and Refinement module) up to the full DWARF architecture. We can notice how the addition of each module always yields better accuracy on most metrics. At first, adding dense connections improves over the baseline model at the cost of nearly triple the number of parameters. By introducing the 3D correlation module, we still improve the capability of the proposed solution to estimate the 3D motion of the scene, this time adding about 2M parameters to the previous 13.5. Finally, adding the Refinement module yields a consistent error reduction on all metrics. Figure 4 shows qualitative results on FlyingThings3D validation set obtained by the full model. More qualitative examples are reported in **supplementary material.**

KITTI 2015. For this experiment, we split the KITTI 2015 training set into 160 images for fine-tuning and reserve the last portion of 40 images for validation purposes only. We report F1, D1 and D2, respectively the percentages of pixels having absolute error larger than 3 and relative error larger than 5% for the three tasks, considering only the non occluded regions (*Noc*) and the whole image (*All*). In this ablation experiments, we aim at assessing the impact of the knowledge distillation protocol on the final accuracy of DWARF. Purposely, we fine-tuned DWARF on two different datasets: i) the 160 images mentioned above from KITTI 2015 and ii) 3200 images from Distilled-KITTI, corresponding to the 160 sequences belonging to KITTI 2015 multi-view extension, respectively reported in the first and the second rows in Table 2. We can notice that proxy labels (Px) yield worse performance for optical flow and consequently for full scene flow while allow improving the accuracy for disparity estimation. Combining the two approaches (i.e., replacing 160 images of the Distilled-KITTI dataset with the real available ground truth, third row) produces result close to using only distilled labels. Finally, running a multi-stage fine-tuning made of 40k steps with proxy labels and further 10k with ground truth (ie, first learning from many yet noisy annotations and then focusing on few perfect labels) dramatically improves the results on optical flow and thus full scene flow, as shown in the fourth row of the table.

4.3 Run-time Analysis

In Table 3, we report the time required to process a couple of stereo images for all variants of DWARF using two different devices. For this purpose, we considered NVIDIA 1080Ti GPU and NVIDIA Jetson TX2, an embedded system equipped with a low-power GPU. The latter device can work with three increasing energy-consumption configurations: *Max-Q* ($<$7.5W), *Max-P* ($\approx$10W) and *Max-N* ($<$15W). Even in its more complex configuration, our network can estimate in the *Max-P* configuration the scene flow on KITTI (4 images padded at 1280×384) in less than 2s, draining about $\frac{1}{25}$ of the energy required by the 1080Ti.

4.4 KITTI 2015 Online Benchmark

Finally, Table 4 reports results for DWARF and state-of-the-art solutions for scene flow, both traditional and based on deep learning. For the final submission, we included all the training data (4000 proxies, 200 ground truths). We followed a 50k (proxy) plus 5k (ground truth) schedule, halv-

Method	D1-All	D2-All	F1-All	SF-All	Params (M)	Runtime (s)
Behl et al. (2017)	4.46	5.95	6.22	8.08	-	600
(Vogel et al. 2015)	4.27	6.79	6.68	8.97	-	300
Ilg et al. (2018)	2.16	6.45	8.60	11.34	116.68	1.72
Ma et al. (2019)	2.55	4.04	4.73	6.31	136.38	1.03
Saxena et al. (2019)	5.13	8.46	12.96	15.69	8.05	0.13
DWARF (ours)	3.33	6.73	10.38	12.78	19.62	0.14

Table 4: Results on the KITTI 2015 online benchmark. Results for (Ilg et al. 2018) from the original paper, since no longer available online. Runtime on NVIDIA 1080 Ti.

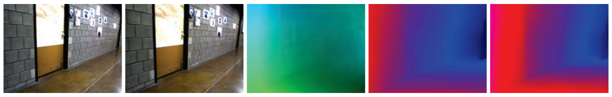

Figure 5: Qualitative results on the WeanHall dataset (Alismail, Browning, and Dias 2011). From left to right, reference frames L_1 and L_2, $\mathcal{F}_1$, $\mathcal{D}_1$ and $\mathcal{D}_{1\leftarrow 2}$.

ing the learning rate at 25K and 35K while reducing it by one quarter at 50K. Despite yielding lower accuracy compared to much more complex state-of-the-art architectures (Ma et al. 2019; Ilg et al. 2018), our network allows us to achieve competitive results using $\sim$ 100M fewer parameters and running more than $10\times$ faster. Compared to approaches closer to ours (Saxena et al. 2019), we can notice that our architecture is much more accurate on all metrics, with margins of about 2.58, 1.8 and 1.73% respectively on F1-All, D1-All and D2-All, leading to a 2.91% improved scene flow estimation, thanks to both 3D correlation layer and knowledge distillation introduced in this paper. Despite counting more than double parameters, DWARF runs almost at the same speed. For a complete comparison with state-of-the-art algorithms, please refer to the KITTI 2015 online benchmark. At the time of writings, DWARF ranks 15^{th}.

4.5 Additional Qualitative Results

We also carried out additional experiments on the WeanHall dataset (Alismail, Browning, and Dias 2011), a collection of indoor stereo images. Since no ground truth is provided, we report qualitative results only. Figure 5 depicts some examples extracted from this dataset processed by the same DWARF model used to submit results to the online KITTI benchmark, proving effective generalization to unseen indoor environments. Finally, we refer the reader to **supplementary material** for additional qualitative results on both synthetic and real datasets, available online.

5 Conclusion

In this paper, we proposed DWARF, a novel and lightweight architecture for accurate scene flow estimation. Instead of combining a stack of task-specialized networks as done by other approaches, our proposal is easily and elegantly trained in an end-to-end fashion to tackle all the tasks at once. Exhaustive experimental results prove that DWARF is competitive with state-of-the-art approaches (Ilg et al. 2018; Ma et al. 2019), counting $6\times$ fewer parameters and running significantly faster. Future work aims at self-adapting DWARF in an online manner (Tonioni et al. 2019b; 2019a).

Acknowledgements. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

Alismail, H.; Browning, B.; and Dias, M. B. 2011. Evaluating pose estimation methods for stereo visual odometry on robots. In *the 11th International Conference on Intelligent Autonomous Systems (IAS-11)*.

Basha, T.; Moses, Y.; and Kiryati, N. 2010. Multi-view scene flow estimation: A view centered variational approach. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1506–1513.

Behl, A.; Jafari, O. H.; Mustikovela, S. K.; Alhaija, H. A.; Rother, C.; and Geiger, A. 2017. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In *International Conference on Computer Vision (ICCV)*.

Butler, D. J.; Wulff, J.; Stanley, G. B.; and Black, M. J. 2012. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV)*.

Chang, J.-R., and Chen, Y.-S. 2018. Pyramid stereo matching network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; and Brox, T. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2758–2766.

Dovesi, P. L.; Poggi, M.; Andraghetti, L.; Martí, M.; Kjellström, H.; Pieropan, A.; and Mattoccia, S. 2019. Real-time semantic stereo matching. *arXiv preprint arXiv:1910.00541*.

Guo, X.; Yang, K.; Yang, W.; Wang, X.; and Li, H. 2019. Group-wise correlation stereo network. In *CVPR*.

He, K.; Gkioxari, G.; Dollar, P.; and Girshick, R. 2017. Mask r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *stat* 1050:9.

Hirschmuller, H. 2005. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, 807–814. IEEE.

Horn, B. K., and Schunck, B. G. 1981. Determining optical flow. *Artificial intelligence* 17(1-3):185–203.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Huguet, F., and Devernay, F. 2007. A variational method for scene flow estimation from stereo sequences. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 1–7. IEEE.

Hui, T.-W.; Tang, X.; and Loy, C. C. 2018. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8981–8989.

Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; and Brox, T. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*, volume 2, 6.

Ilg, E.; Saikia, T.; Keuper, M.; and Brox, T. 2018. Occlusions, motion and depth boundaries with a generic network for optical flow, disparity, or scene flow estimation. In *15th European Conference on Computer Vision (ECCV)*.

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

Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.; Kennedy, R.; Bachrach, A.; and Bry, A. 2017. End-to-end learning of geometry and context for deep stereo regression. In *The IEEE International Conference on Computer Vision (ICCV)*.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Liang, Z.; Feng, Y.; Guo, Y.; Liu, H.; Chen, W.; Qiao, L.; Zhou, L.; and Zhang, J. 2018. Learning for disparity estimation through feature constancy. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, P.; King, I.; Lyu, M.; and Xu, J. 2019. Ddflow: Learning optical flow with unlabeled data distillation. In *AAAI Conference on Artificial Intelligence*.

Luo, W.; Schwing, A. G.; and Urtasun, R. 2016. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5695–5703.

Ma, W.-C.; Wang, S.; Hu, R.; Xiong, Y.; and Urtasun, R. 2019. Deep rigid instance scene flow. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4040–4048.

Meister, S.; Hur, J.; and Roth, S. 2018. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Menze, M., and Geiger, A. 2015. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Menze, M.; Heipke, C.; and Geiger, A. 2015. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*.

Poggi, M.; Pallotti, D.; Tosi, F.; and Mattoccia, S. 2019. Guided stereo matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pons, J.-P.; Keriven, R.; and Faugeras, O. 2007. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *Int. J. Comput. Vision* 72(2):179–193.

Ranjan, A., and Black, M. J. 2017. Optical flow estimation using a spatial pyramid network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Revaud, J.; Weinzaepfel, P.; Harchaoui, Z.; and Schmid, C. 2015. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *Computer Vision and Pattern Recognition*.

Saxena, R.; Schuster, R.; Wasenmüller, O.; and Stricker, D. 2019. PWOC-3D: Deep occlusion-aware end-to-end scene flow estimation. In *Intelligent Vehicles Symposium (IV)*.

Song, X.; Zhao, X.; Hu, H.; and Fang, L. 2018. Edgestereo: A context integrated residual pyramid network for stereo matching. In *Asian Conference on Computer Vision (ACCV)*.

Sun, D.; Roth, S.; Lewis, J. P.; and Black, M. J. 2008. Learning optical flow. In Forsyth, D.; Torr, P.; and Zisserman, A., eds., *Computer Vision – ECCV 2008*, 83–97. Berlin, Heidelberg: Springer Berlin Heidelberg.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2017. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. *ArXiv e-prints*.

Sun, D.; Roth, S.; and Black, M. J. 2014. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision* 106(2):115–137.

Taniai, T.; Sinha, S. N.; and Sato, Y. 2017. Fast multi-frame stereo scene flow with motion segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tonioni, A.; Rahnama, O.; Joy, T.; Stefano, L. D.; Ajanthan, T.; and Torr, P. H. 2019a. Learning to adapt for stereo. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tonioni, A.; Tosi, F.; Poggi, M.; Mattoccia, S.; and Di Stefano, L. 2019b. Real-time self-adaptive deep stereo. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Valgaerts, L.; Bruhn, A.; Zimmer, H.; Weickert, J.; Stoll, C.; and Theobalt, C. 2010. Joint estimation of motion, structure and geometry from stereo sequences. *Computer Vision–ECCV 2010* 568–581.

Vedula, S.; Baker, S.; Rander, P.; Collins, R.; and Kanade, T. 1999. Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, 722–729 vol.2.

Vogel, C.; Schindler, K.; and Roth, S. 2011. 3d scene flow estimation with a rigid motion prior. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 1291–1298. IEEE.

Vogel, C.; Schindler, K.; and Roth, S. 2013. Piecewise rigid scene flow. In *Proceedings of the IEEE International Conference on Computer Vision*, 1377–1384.

Vogel, C.; Schindler, K.; and Roth, S. 2015. 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision* 115(1):1–28.

Wedel, A.; Brox, T.; Vaudrey, T.; Rabe, C.; Franke, U.; and Cremers, D. 2011. Stereoscopic scene flow computation for 3d motion understanding. *International Journal of Computer Vision* 95(1):29–51.

Wulff, J., and Black, M. J. 2015. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 120–130.

Yang, G.; Zhao, H.; Shi, J.; Deng, Z.; and Jia, J. 2018. Segstereo: Exploiting semantic information for disparity estimation. In *15th European Conference on Computer Vision (ECCV)*.

Yin, Z.; Darrell, T.; and Yu, F. 2019. Hierarchical discrete distribution decomposition for match density estimation. In *CVPR*.

Zbontar, J., and LeCun, Y. 2016. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research* 17(1-32):2.

Zhang, Y., and Kambhamettu, C. 2001. On 3d scene flow and structure estimation. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, II–778–II–785 vol.2.

Zhang, F.; Prisacariu, V.; Yang, R.; and Torr, P. H. 2019. Ga-net: Guided aggregation net for end-to-end stereo matching. In *CVPR*.