Fast Stereo Disparity Maps Refinement By Fusion of Data-Based And Model-Based Estimations

Maxime Ferrera, Alexandre Boulch, Julien Moras

Fast Stereo Disparity Maps Refinement By Fusion of Data-Based And Model-Based Estimations

Maxime Ferrera[1,2] Alexandre Boulch[1] Julien Moras[1]

[1]DTIS, ONERA, Université Paris Saclay F-91123 Palaiseau, France
[2]LIRMM, Université Montpellier, CNRS, Montpellier, France
first_name.last_name@onera.fr

Abstract

The estimation of disparity maps from stereo pairs has many applications in robotics and autonomous driving. Stereo matching has first been solved using model-based approaches, with real-time considerations for some, but today's most recent works rely on deep convolutional neural networks and mainly focus on accuracy at the expense of computing time. In this paper, we present a new method for disparity maps estimation getting the best of both worlds: the accuracy of data-based methods and the speed of fast model-based ones. The proposed approach fuses prior disparity maps to estimate a refined version. The core of this fusion pipeline is a convolutional neural network that leverages dilated convolutions for fast context aggregation without spatial resolution loss. The resulting architecture is both very effective for the task of refining and fusing prior disparity maps and very light, allowing our fusion pipeline to produce disparity maps at rates up to 125 Hz. We obtain state-of-the-art results in terms of speed and accuracy on the KITTI benchmarks. Code and pre-trained models are available on our github: https://github.com/ferreram/FD-Fusion.

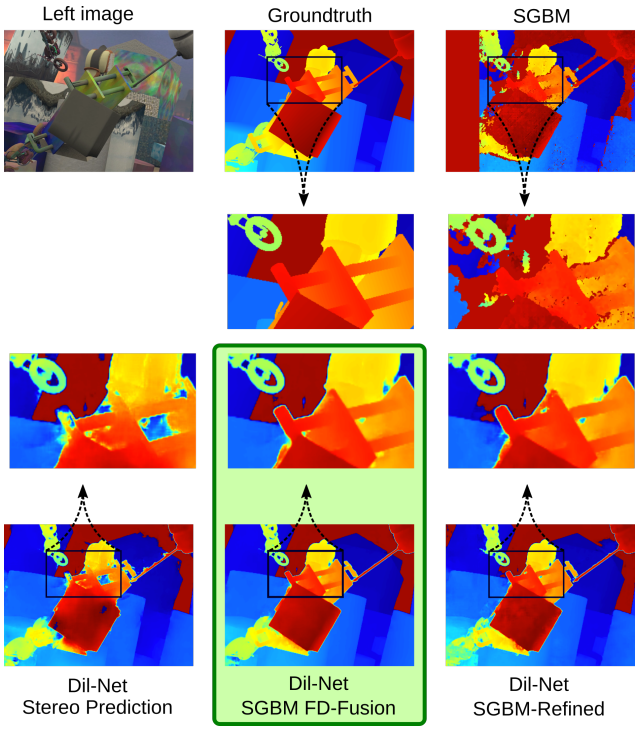

Figure 1. FD-Fusion pipeline results on a sample image from SceneFlow. Top row: left stereo image (left), groundtruth disparity map (middle), SGBM estimate (right). Middle rows: zoom-in on the disparity maps. Last row: Dil-Net stereo-only estimate (left), FD-Fusion output (middle), SGBM refined with Dil-Net (right).

1. Introduction

In mobile robotics and autonomous driving, knowledge about the 3D structure of the environment is required for safe navigation. This 3D information has to be inferred from embedded sensors such as LiDAR, RGB-D cameras or stereo cameras. Stereo setups offer the great advantage of working both indoors and outdoors, in opposition to RGB-D cameras which are limited to indoor environments. Besides, they are cheaper and lighter than LiDAR systems, making them easier to use and embed on weight-constrained robots such as unmanned aerial vehicles (UAV).

As the 3D information is used for decision-making in autonomous navigation, highly accurate 3D measurements should be produced at high-rates. Indeed, safety in such context depends on the accuracy of the 3D measurements and on the frequency of these measurements, both directly impacting the pertinence of the given measurements. These speed constraints have led to the development of several fast algorithms for disparity maps estimation from stereo images and, today, hardware-specific systems output disparity

map at cameras' acquisition rate (60 Hz for Mynt-Eye cameras[1], 90 Hz for Intel RealSense[2]). However, such maps are estimated from model-based algorithms and lack the accuracy of current state-of-the-art methods relying on Convolutional Neural Networks (CNN).

In this paper, we tackle the challenge of producing accurate disparity maps from stereo images while respecting the timing constraints required for safe autonomous navigation. We propose a fusion pipeline that combines disparity maps produced by model-based and data-based stereo matching methods, acting as a fast disparity refinement filter. The proposed pipeline is built on-top of a light CNN that leverages dilated convolutions [25] (also referred to as *atrous* convolutions) in order to quickly increase the receptive field of the CNN without giving up on spatial resolution. This allows to aggregate context information in a fast way to correct the disparity estimated at each pixel.

The developed CNN architecture is highly versatile as it can be used for disparity maps prediction from stereo images only, refine prior disparity maps and fuse model-based and learning-based stereo matching methods outputs, highly increasing the accuracy of the inferred maps. It is also extremely fast as a single-pass through the network takes an average of 2.5 ms on GPU. This CNN being the core of the proposed fusion pipeline, we hence propose a new method able to predict accurate disparity maps with a minimal computation time.

We list the paper contributions as follows:

- Dil-Net: a light CNN architecture, based on dilated convolutions, able to combine and refine prior disparity maps and predict disparity maps from stereo images
- FD-Fusion, for Fast-Disparity-Fusion: a new pipeline for fusing model-based and data-based disparity maps and produce a refined result
- We show that combining model-based and data-based disparity maps largely increase the accuracy of the final results, highlighting the fact that complementary features exist between both kind of approaches

The paper is organized as follow: section 2 presents the works related to our approach, section 3 details our pipeline for fast disparity maps refinement and fusion and presents the detail of Dil-Net's architecture. Finally, section 4 exposes the implementation details of our method, an ablation study of the main components of the pipeline and compare the proposed approach to state-of-the-art methods on the KITTI online benchmarks.

2. Related Work

Model-based Stereo Matching. In the past decades, the stereo matching problem has been widely studied and mainly solved using model-based methods [10]. These methods mainly rely on semi-local [12, 7] or global matching [15] to infer disparity maps based on stereo images, usually rectified to satisfy epipolar geometry constraints. The great advantage of these methods is that they have been developed on-top of photometric and geometric heuristics, making them quite reliable for most kind of scenes.

Moreover, many have been developed for robotic applications and are hence computationally quite fast, even on CPUs. However, as matching texture-less areas is quite problematic in computer vision, these methods tend to produce false results in such areas. Fine details, such as small objects and objects boundaries, are also difficult to recover with such methods. By using the products from these methods as an input for our refinement approach, we intend to benefit from their robustness and computation speed.

CNN-based Stereo Matching. Since [26], the use of convolutional neural networks has been highly investigated to solve stereo matching. First works relied on CNNs to assess the matching of stereo image patches. While giving more accurate results than model-based methods, the matching process was highly ineffective in terms of computational load (67s per image on GPU). The authors of [16] designed DispNet, the first end-to-end architecture able to estimate disparity maps from stereo pairs of images. Following this seminal work, [13] proposed an end-to-end architecture inspired by the classical stereo matching pipeline. Many works followed this trend [2, 23, 20, 18, 24, 9, 5] and completely outmatched previous methods in terms of accuracy. However, this gain in accuracy is at the cost of high computation requirement. For real-time applications such as robotics or autonomous driving, disparity maps are required at a higher-rate in order to detect as quickly as possible potential obstacles or dangers and make decision upon it.

A few works take this run-time constraint into account and proposed CNN-based methods able to process stereo images at high-rate. First, [14] proposed StereoNet, the first CNN architecture with a high frame-rate suitable for real-time applications. Following this work, [22] presented a CNN not only able to produce disparity maps at high-rate but also able to learn online to refine its predictions. Nonetheless, there is a trade-off between speed and accuracy. With respect to these methods, we intend to lower this trade-off by keeping a very high frame-rate while increasing the accuracy.

Disparity Maps Enhancement. Some works also tackled the challenge of refining disparity maps. The authors of [8], proposed to refine the outputs of a given baseline for dense pixel-wise labelling tasks such as semantic segmentation or disparity maps regression. Focusing on the refinement of disparity maps, [1] designed a deep architecture relying on recurrent neural networks and [3, 4] proposed a

[1] https://www.mynteye.com/products/mynt-eye-stereo-camera

[2] https://www.intelrealsense.com/

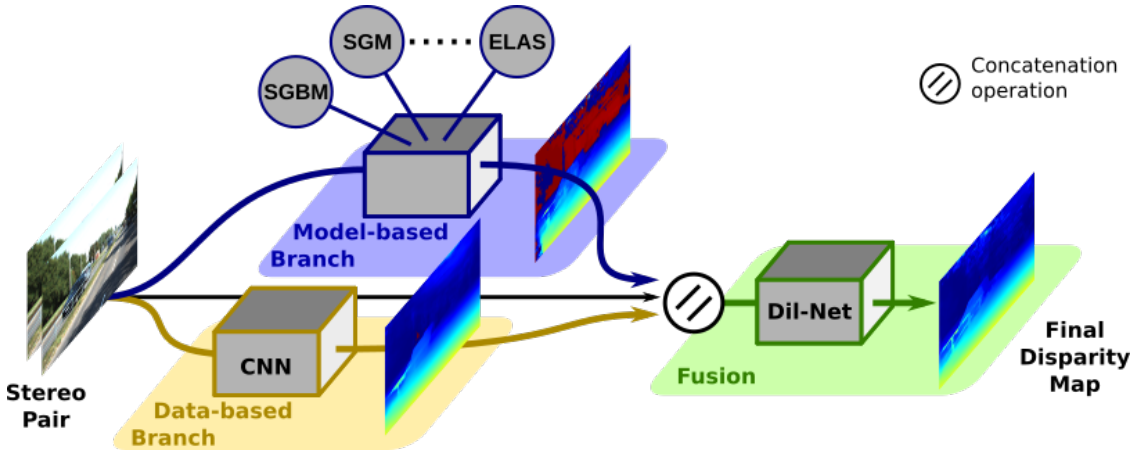

Figure 2. The FD-Fusion pipeline. The top of the pipeline is composed of a data-based branch and a model-based one, each taking a stereo pair of images as input and estimating a disparity map from it. The next stage is the fusion pipeline, an instance of Dil-Net, which takes as inputs the resulting disparity maps as well as the original stereo images to produce a refined disparity map by fusing the prior ones.

spatial propagation network. In opposition to the proposed approach, these methods rely only on data-based disparity maps refinement and do not take into account any timing constraint. Closer to our work, [19] investigated the fusion of model-based disparity maps produced by different methods. Their method take as input the disparity hypothesis of eight different methods and choose the best candidate for each pixel. In our approach, we also operate a fusion of multiple disparity maps. However, we use both model-based and data-based outputs to perform the fusion. Furthermore, their method is only meant for fusing disparity maps whereas in our approach a refinement of individual disparity maps is also possible.

3. The FD-Fusion Pipeline

The proposed pipeline for disparity maps estimation is presented on Figure 2. It is a two-stage pipeline. First, initial disparity maps are computed on two parallel branches: one using a model-based approach, the other using a data based approach, typically a neural network. Second, the generated maps are then stacked with the stereo pairs and given to the fusion module that will merge the input information to generate an enhanced disparity map. The general assumption is that, first both model-based and data-based estimations are sensible to different features of the input image, and second, their failure cases are also different. In this case, the objective is for the fusion module to benefit from both model-based and learned disparity maps while refining the borders according to the original stereo pair.

3.1. Fusion module

The fusion module (in green in Figure 2) aims at exploiting the input maps and images to produce a disparity map. In this work, we learn this fusion step using a convolution neural network.

The proposed method is based on the use of dilated convolutions [25] inside a small CNN in order to refine disparity maps. The idea is to learn how a disparity map estimated by a given method should be corrected to be closer to reality. By using dilated convolutions, we quickly increase the receptive field of the deep network without decreasing too much the input resolutions. This allows the aggregation of context information around every pixel. This information will be used as a clue to predict whether a disparity value is good or not and what value is most likely to be the correct one given the surrounding of the pixel. Hence, by feeding the network with a stereo pair along with the disparity map estimated from it, we want to infer a new disparity map which takes into account the disparity values and the visual context. By concatenating the inputs along one dimension, we avoid the need of several input branches in the network and hence limit the number of parameters and reduce the processing time.

Disparity maps predicted by learning-based methods are going to use very different heuristics than model-based methods. By concatenating the disparity maps estimated from learning-based and model-based methods, along with their respective stereo pairs, the network can then learn how to combine them by analyzing the strength and weakness of each methods given a visual context. By doing so, the network can infer refined disparity maps that surpass the ones that would have been estimated separately (*i.e.* the one estimated by refining only the disparity map computed by method 1 or 2).

The strength of the proposed network is that it can perform either of these tasks in an extremely fast way. Indeed, a single pass through Dil-Net takes an average of 2.5 ms, making it compatible with real-time requirements such as in robotics.

Name	Layer Settings	Output Dimension
Input		$H \times W \times N_{in}$
CNN		
conv_1	k=3, s=1, d=1	$H \times W \times 64$
conv_2	k=3, s=2, d=1	$\frac{1}{2}H \times \frac{1}{2}W \times 128$
Dilated Part		
conv_3	k=3, s=1, d=2	$\frac{1}{2}H \times \frac{1}{2}W \times 128$
conv_4	k=3, s=1, d=4	$\frac{1}{2}H \times \frac{1}{2}W \times 128$
conv_5	k=3, s=1, d=8	$\frac{1}{2}H \times \frac{1}{2}W \times 128$
conv_6	k=3, s=1, d=12	$\frac{1}{2}H \times \frac{1}{2}W \times 128$
conv_7	k=3, s=1, d=8	$\frac{1}{2}H \times \frac{1}{2}W \times 128$
conv_8	k=3, s=1, d=4	$\frac{1}{2}H \times \frac{1}{2}W \times 128$
conv_9	k=3, s=1, d=2	$\frac{1}{2}H \times \frac{1}{2}W \times 128$
End of Dilated Part		
conv_10	k=3, s=2, d=1	$H \times W \times 128$ concat with conv_1 output
conv_11	k=3, s=1, d=1	$H \times W \times 128$ concat with input
conv_12	k=3, s=1, d=1	$H \times W \times 128$
conv_13	k=3, s=1, d=1	$H \times W \times 128$
conv_14	k=3, s=1, d=1	$H \times W \times 64$
conv_15	k=3, s=1, d=1	$H \times W \times 32$
conv_16	k=1, s=1, d=1	$H \times W \times N_{out}$

Table 1. Architecture of the proposed network: Dil-Net. Symbols meaning: k is the convolution kernel size, s is the stride factor (down or up), d is the dilatation rate and N_{in}/N_{out} are the dimensions of the input / output (in terms of channels).

Network Description The network structure, referred to as *Dil-Net*, is presented in table 1. First the input is pre-processed through two convolutional layers for extracting low-level features. The second layer also reduces the resolution of the input by a factor of 2 to start increasing the receptive field of the network. Then the feature maps are processed through a total of seven convolution layers composed of filters with first increasing and then decreasing dilatation rates. This dilated part acts has a context aggregation pipeline with a bigger and bigger receptive field. The output of this section is then up-sampled and concatenated to its input feature maps by means of a skip-connection. In the next layer, we further add the input of the network to the current features map. The following four layers are then used for merging step by step the feature maps into a 32-D feature maps which is finally used to infer a disparity value for each pixel through a convolution kernel of size 1.

3.2. Model-based branch

The model-based branch (in blue in Figure 2) produce disparity image using classical handcrafted stereo matching algorithms. We have experimented three different algorithms: SGBM [12], SGM [11] and ELAS [7].

The first kind of method tested is SGBM / SGM which are semi-global approaches that compute a matching score for each disparity hypotheses (computed on a single pixel for SGM and on a patch for SGBM). This score is aggregated to compute cost that considers the cost of neighboring points (approximating this using a finite set of directions) and disparities. One main issue with SGBM is that the block matching is not possible on the extreme side of the image leaving a band where the disparity is not computed.

The second method considered is ELAS. It is a semi-local approach that performs a two-stage estimation. First, it generates a prior disparity map by computing a robust disparity over a sparse grid and next extend the disparity to the whole image by using a triangulation over this set of support points. Then a bayesian inference scheme is used to fuse this prior and image similarity score and to compute final disparity map.

3.3. Data-based branch

The data-based branch (in yellow in Figure 2) uses a learning algorithm to infer a disparity map from a stereo image. The major difference with the model-based approach is that these models are trained using supervised learning, *i.e.* giving the stereo image and the groundtruth disparity map during training phase. Once the network have been trained, it can be used to infer disparity maps from new stereo images.

3.4. Training

The loss used to train the network is a ℓ_1-loss:

$$r(x) = \frac{1}{N} \sum_{n}^{N} |x_n - y_n| \tag{1}$$

where $r(x)$ is the residual, N is the number of samples, y_n is the groundtruth and x_n is the output of the network.

This architecture is based on only 16 layers and simply relies on 3×3 convolution kernels (except for the last layer), making it extremely efficient in terms of computation.

The FD-Fusion pipeline is a multi-stage structure. However, the training of the CNN-based parts of the pipeline is not done end-to-end. First the data-based branch is trained for its own task, *i.e.* stereo-only predictions, and then the fusion module is trained in a second step.

3.5. Modularity of the Pipeline

Despite the speed and lightness of Dil-Net, the proposed network is highly versatile in its use and can be used to perform the following tasks:

- Predict disparity maps from stereo images only
- Refine single disparity map
- Combine multiple disparity maps to produce a refined result

Furthermore, the proposed FD-Fusion pipeline is very modular in that it can take any method for the data-based and model-based branches. The number of branches can

even be adjusted as needed, by adding more model-based methods and removing the data-based branch for example. The final stage of the pipeline, an instance of Dil-Net, is then responsible of fusing all the prior disparity maps.

In this paper, as we focus on fast disparity map estimations, we stick to the Dil-Net architecture for the data-based branch of the pipeline and to model-based methods relying on local matching. In the following, when we refer to FD-Fusion, we are assuming that a stereo-only instance of Dil-Net is used as the data-based module.

4. Experiments

We have evaluated the proposed pipeline on three stereo datasets: SceneFlow [16], KITTI 2012 [6] and KITTI 2015 [17]. We first describe the datasets used and the implementation details of our method. An ablation study is presented on SceneFlow and KITTI 2015 and then follow the evaluation of our best combinations on the official KITTI 2012 and 2015 benchmarks.

4.1. Datasets description

The stereo datasets used for the evaluation are:

1. **Scene Flow:** a large synthetic dataset providing stereo RGB pairs along with their dense disparity map groundtruth. The dataset contains 35454 training and 4370 testing images of size 960×540.
2. **KITTI 2015:** a real-world dataset focusing on autonomous driving for cars. It provides stereo RGB pairs as well as their associated disparity map in the form of a sparse groundtruth extracted from an embedded LiDAR. The training and testing split are of 200 images each with an average size of 1240×376. The groundtruth is only available for the training split. In order to validate our models, we have further divided the training split into a training set of 160 images (80%) and a validation set of 40 images (20%).
3. **KITTI 2012:** another dataset focusing on autonomous driving for cars, similar to KITTI 2015.

4.2. Implementation Details

The architecture details of Dil-Net are given in Table 1, that is the size of the convolution layers, the dilatation rate, the activation policy and the resolution of the features map.

The network has been implemented using Pytorch. The different Dil-Net instances tested are the following: a stereo-only model, a refinement model (not using the data-based branch) per model-based method and fusion models that combine model-based outputs with the stereo-only Dil-Net predictions. Moreover, we have trained Dil-Net instances taking as input the model-based methods estimations altogether. All these models have been trained using the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$). The RGB images provided as inputs are normalized by dividing them by 255. The prior disparity maps and the groundtruth maps are converted into approximate *z-scores* maps ($disp = \frac{disp-\mu}{\sigma}$) — taking μ and σ as the mean and standard deviation of the datasets disparity maps groundtruth. The inputs fed to the network are resized to 512×256. The outputs of the network are also of size 512×256 and are hence up-sampled afterwards through bilinear interpolation to match the original image size before computing the error metrics.

All the models are first trained on SceneFlow for 30 epochs, halving the learning rate every 5 epochs on the first 20 epochs and then every 2 epochs. The pre-trained models are then separately fine-tuned on each one of the KITTI set for 500 epochs, halving the learning rate every 50 epochs. On an NVIDIA GTX 1080-Ti, the training of one model on SceneFlow takes an average of 30 hours and around 3 hours on each KITTI sets. The inference time is around 2.5 ms for a one-pass through the network and around 4.5 ms when we combine two instances of this network (when using one instance to predict a disparity map given a stereo pair of images and another instance to fuse this output with the output of a model-based method for example).

When training on SceneFlow, for the models trained for the tasks of disparity map prediction from stereo images and of model-based disparity maps refinement, the starting learning rate is 0.001. For the ones used for model-based and data-based outputs fusion, the starting learning rate is 0.002. When finetuning the different models on the KITTI sets, the starting learning rates are respectively divided by 2 given the ones used on SceneFlow. In all the different trainings, the batch size is set by the maximum size allowed by the GPU (typically between 4 and 6 depending on the models on a GTX 1080-Ti).

On every dataset, the inputs are randomly flipped horizontally and, on the KITTI sets, the input RGB images are further modified by randomly modifying the level of brightness, contrast, hue and saturation.

The tested model-based methods are the following: SGBM [12], ELAS [7] and a CUDA-version of SGM [11] (referred to as Cuda-SGM). For SGBM, we use the OpenCV implementation of the algorithm[3]. For ELAS, we use the python wrapper of the algorithm[4]. For Cuda-SGM, we use the code available on their github[5]. The different methods' parameters used in this work are detailed on our github repository.

4.3. Ablation Study

The objective of this section is to motivate each block of the proposed pipeline and to validate the efficiency of Dil-Net. Results are presented in table 2 and table 5.

[3] https://docs.opencv.org/3.4.2/d2/d85/classcv_1_1StereoSGBM.html

[4] https://github.com/jlowenz/pyELAS

[5] https://github.com/dhernandez0/sgm

Methods	No refinement	Refinement	FD-Fusion
Cuda-SGM [11]	7.56 px	2.09 px	1.57 px
ELAS [7]	4.41 px	2.37 px	1.54 px
SGBM [12]	3.04 px	1.92 px	1.51 px
Multi-models	-	1.63 px	1.37 px
Dil-Net	2.83 px	-	-

Table 2. Absolute mean error (End-Point-Error) on the SceneFlow test set.

Methods	No refinement	Refinement	FD-Fusion
Cuda-SGM [11]	8.11 %	4.13 %	3.07 %
ELAS [7]	10.28 %	5.84 %	3.96 %
SGBM [12]	5.15 %	4.58 %	3.19 %
Multi-models	-	3.84 %	3.03 %
Dil-Net	8.69 %	-	-

Table 3. Percentage of pixels misclassified (error $>$ 3 px) on the KITTI 2015 validation set (40 images).

4.3.1 Results on SceneFlow

We now compare the performance of each module of the proposed FD-Fusion pipeline. We first evaluate on the SceneFlow test set. Following [13, 2], we do not take into account pixels with disparity value higher that 192 when computing the loss at training time and when computing the error metrics at testing time. The metric used for evaluation is the End-Point-Error (EPE), that is the mean of the absolute error at each pixel. The results obtained are listed in Table 2. We give the results of the raw model-based methods and of Dil-Net trained with stereo images only in the *no refinement* column. Then we compare the accuracy obtained after refinement of each model-based methods taken individually and taken altogether (Multi-models raw in the table). We finally compare the final results obtained with our FD-Fusion scheme, using the output of the stereo-only Dil-Net as the data-based module. First, the *refinement* column results assess the refining abilities of the Dil-Net architecture, increasing the accuracy of all the methods by a large rate (72.5 % with Cuda-SGM, 42 % with ELAS, 37 % with SGBM). We also note that, while the results of stereo-only Dil-Net are finer on this dataset than the ones of the model-based methods, we get more accurate results after refinement of the prior disparity maps. Besides, the network produces even better results when taking the multiple models outputs as inputs, highlighting its efficiency in fusing disparity maps estimated by different means. Secondly, when evaluating the full FD-Fusion pipeline outputs we see an even bigger gain in terms of accuracy (80 % with Cuda-SGM, 62 % with ELAS, 50 % with SGBM and 16 % over the fused and refined multi-models). Given these last results, we demonstrate that data-based produced disparity maps and model-based ones have complementary features that can be efficiently combined to produce more accurate results. We also assess the efficiency of the Dil-Net architecture for this task in that it fully leverages all the given inputs to estimate a refined results (see Figure 1).

4.3.2 Results on KITTI

We further compare the performance of the proposed pipeline on the KITTI split, created by taking 160 images for training and 40 for testing from the training split of the KITTI 2015 official set. The metric used for evaluation on this set is the percentage of pixels misclassified (error $>$ 3 px). The evaluation of each method is done after finetuning the different instances of Dil-Net pre-trained on SceneFlow. The obtained results are listed in Table 3. First, we can note that on this dataset the stereo-only Dil-Net is outperformed by all the model-based methods but Cuda-SGM. Second, we see that the pattern of the results follows the one obtained on the SceneFlow dataset. The refinement of each model-based outputs largely surpasses the prior estimations (49 % with Cuda-SGM, 43 % with ELAS, 11 % with SGBM). Once again, the results obtained with the model-based estimations taken altogether are better than their indivual refinement. Furthermore, we see that the FD-Fusion scheme also produces finer results (62 % with Cuda-SGM, 61 % with ELAS, 38 % with SGBM and 21 % over the fused and refined multi-models). Given the poorer accuracy of the stereo-only Dil-Net estimations on this dataset, we highlight the fact that data-based on model-based outputs contain very different features that can be reliably extracted and combined by CNN architectures. Qualitative results are given in Figure 3.

4.3.3 Results Analysis

Given all these results, we show that the proposed approach offers a big boost in terms of accuracy with respect to its processing time (2.5 ms for a single-pass through one instance of Dil-Net). We compare the run-time performance of the different methods in Table 4 on their respective hardware implementations. As it can be seen, the main bottleneck of the FD-Fusion pipeline in terms of speed comes from the on-top branches (data-based and model-based modules). If we take Cuda-SGM as the model-based module and an instance of Dil-Net as the data-based one, we can reach rates of around 125 Hz for the estimation of refined disparity maps (3.5 ms for the prior estimation of a disparity map with Cuda-SGM and 4.5 ms for the inference of a disparity map from stereo images with Dil-Net and the fusion and refinement of both through another instance of Dil-Net). Note that, this calculus is done by taking the different step sequentially. However, one could assume that the model-based and data-based outputs are computed in parallel and then concatenated and fed to the fusion module, in which case we could reach rates of approximately 165 Hz (3.5 ms + 2.5 ms per disparity maps).

We also stress out the fact that while we have only used

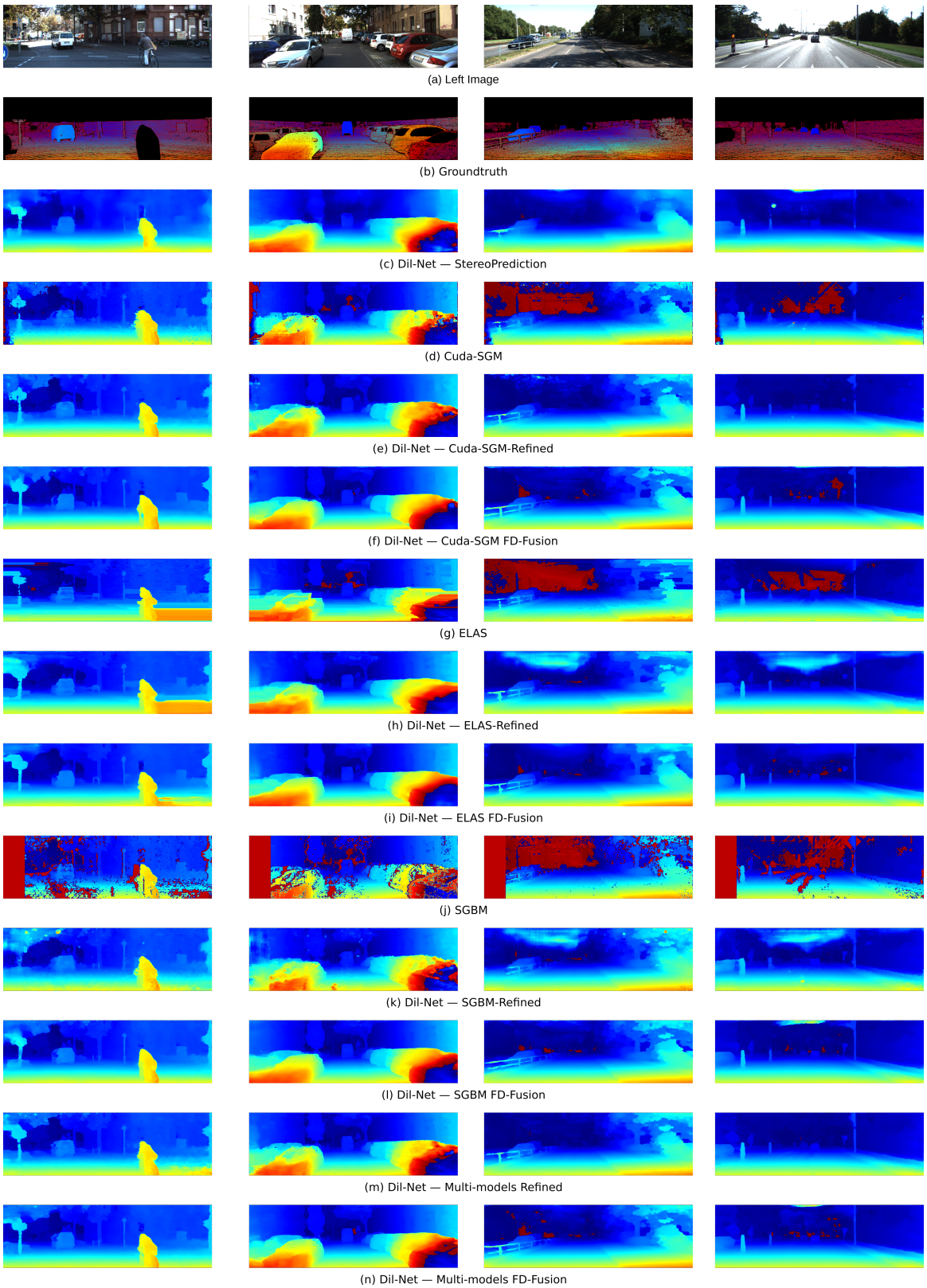

Figure 3. Result samples from our KITTI 2015 validation set. (a) Left stereo image, (b) groundtruth disparity, outputs of (c) stereo-only Dil-Net, (d) Cuda-SGM, (e) Cuda-SGM refined , (f) FD-Fusion with Cuda-SGM, (g) ELAS, (h) ELAS refined, (i) FD-Fusion with ELAS, (j) SGBM, (k) SGBM refined, (l) FD-Fusion with SGBM, (m) the multi-models refined, (n) FD-Fusion with the multi-models.

Methods	SGBM	ELAS	Cuda-SGM	Dil-Net (single-pass)	Dil-Net (two-passes)
Hardware	Intel Xeon 3.0GHz	Intel Xeon 3.0GHz	GTX-1070 / GTX 1080-Ti	GTX 1070 / GTX 1080-Ti	GTX 1070 / GTX 1080-Ti
Runtime	310 ms	210 ms	7.5 / 3.5 ms	2.5 / 2.5 ms	4.5 / 4.5 ms

Table 4. Run-time analysis of the different stereo matching methods given the tested harware. Dil-Net two-passes denotes the total processing time when loading two instances of Dil-Net and feeding the second one with the ouput of the first one.

	KITTI Stereo 2015			**KITTI Stereo 2012 (< 2 px / < 3 px)**				
Method	**D1-bg**	**D1-fg**	**D1-all**	**Out-Noc**	**Out-All**	**Avg-Noc**	**Avg-All**	**Runtime**
Precision-oriented methods								
M2S_CSPN [4]	**1.51 %**	**2.88 %**	**1.74 %**	**1.79 / 1.19 %**	**2.27 / 1.53 %**	**0.4** px	**0.5 px**	500 ms
AMNet [5]	1.53 %	3.43 %	1.84 %	2.12 / 1.32 %	2.71 / 1.73 %	0.5 px	**0.5 px**	900 ms
MS_CSPN [4]	1.56 %	3.78 %	1.93 %	-	-	-	-	500 ms
GANet-15 [27]	1.55 %	3.82 %	1.93 %	2.18 / 1.36 %	2.79 / 1.80 %	0.5 px	**0.5 px**	360 ms
HD^3-Stereo [24]	1.70 %	3.63 %	2.02 %	2.00 / 1.40 %	2.56 / 1.80 %	0.5 px	**0.5 px**	**140 ms**
EdgeStereo-V2 [21]	1.84 %	3.30 %	2.08 %	2.32 / 1.46 %	2.88 / 1.83 %	**0.4 px**	**0.5 px**	320 ms
ECMUA [18]	1.66 %	4.27 %	2.09 %	2.02 / 1.26 %	2.56 / 1.64 %	**0.4 px**	**0.5 px**	900 ms
GwcNet-g [9]	1.74 %	3.93 %	2.11 %	2.16 / 1.32 %	2.71 / 1.70 %	0.5 px	**0.5 px**	320 ms
EdgeStereo [20]	1.87 %	3.61 %	2.16 %	-	-	-	-	700 ms
SegStereo [23]	1.88 %	4.07 %	2.25 %	2.66 / 1.68 %	3.19 / 2.03 %	0.5 px	0.6 px	600 ms
PSMNet [2]	1.86 %	4.62 %	2.32 %	2.44 / 1.49 %	3.01 / 1.89 %	0.5 px	0.6 px	410 ms
Running time oriented methods								
DispNetC [16]	4.32 %	**4.41 %**	4.34 %	7.38 / 4.11 %	8.11 / 4.65 %	0.9 px	1.0 px	60 ms
MADnet [22]	3.75 %	9.20 %	4.66 %	-	-	-	-	20 ms
StereoNet [14]	4.30 %	7.45 %	4.83 %	4.91 / - %	6.02 / - %	0.8 px	0.9 px	15 ms
FD-Fusion with Cuda-SGM	**3.22 %**	7.44 %	**3.92 %**	**4.8 / 3.16 %**	**5.73 / 3.85 %**	**0.7 px**	**0.8 px**	**8 ms**

Table 5. KITTI Stereo 2015 - 2012 - Official benchmark results.

Dil-Net as the data-based module in these experiments to fulfill the run-time constraints of real-time applications, any other method could be used as a replacement. One should note that we do not process images at full resolution to speed-up the inference time here. Hence, adding a very accurate method as the data-based module might require to adapt the fusion module in order to see improvements as the Dil-Net architecture has been designed for low-resolution inputs. These considerations are left for future work.

4.4. KITTI Stereo 2015 / 2012

We evaluate our pipeline against state-of-the-art methods on the KITTI Stereo 2012 and 2015 official benchmarks. On the KITTI Stereo 2015 benchmark, the evaluation metric is the percentage of misclassified pixels (error $>$ 3 px) and is given for the foreground (D_1-fg) and background (D_1-bg) part of the images as well as for the whole image (D_1-all). On the KITTI Stereo 2012 benchmark, the evaluation metrics are the percentage of misclassified pixels given different threshold (we give the results for error $>$ 2 and $>$ 3 px here) and the average disparity error. Furthermore, these results are computed for the non-occluded part of the images and the whole images (Noc / All).

The disparity maps submitted to the online platform are the ones resulting from the use of Cuda-SGM as the model-based module within the FD-Fusion pipeline, *i.e.* the fastest combination. The obtained results are given in Table 5. We give both the results of *precision-oriented* and *run-time-oriented* methods in this table to give a good overview of the current state-of-the-art methods in terms of speed and accuracy on the KITTI Stereo benchmarks. In the category of fast methods we find – ranked by speed – DispNet [16], MADnet [22] and StereoNet [14]. Compared to these methods, our proposed pipeline is the fastest, being almost two times faster than StereoNet. Besides, we also obtain the best performance in terms of accuracy, surpassing all the three methods on every metric but the one considering the foreground part of the image in the KITTI 2015 benchmark, on which only DispNet performs better.

5. Conclusion

In this paper we have presented FD-Fusion, a new pipeline for fast disparity maps refinement through the fusion of model-based and data-based outputs. The core of this pipeline is Dil-Net, a CNN based on dilated convolutions that is able to predict, refine and fuse disparity maps to produce accurate results at very high rates. The full pipeline can be used to produce state-of-the-art disparity maps at rates up to 125 Hz. We have shown that one of the strength of the proposed pipeline lies in the fusion of data-based and model-based methods which exhibits complementary features. Future works will include investigating the use of different data-based methods at the top of the pipeline. We will also investigate the potential effectiveness of the proposed pipeline for other dense regression tasks such as monocular depth estimation or dense optical flow.

References

[1] K. Batsos and P. Mordohai. RecResNet: A recurrent residual cnn architecture for disparity map enhancement. *2018 International Conference on 3D Vision (3DV)*, pages 238–247, 2018. 4322

[2] J.-R. Chang and Y.-S. Chen. Pyramid stereo matching network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 4322, 4326, 4328

[3] X. Cheng, P. Wang, and R. Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018. 4322

[4] X. Cheng, P. Wang, and R. Yang. Learning depth with convolutional spatial propagation network. *arXiv preprint arXiv:1810.02695*, 2018. 4322, 4328

[5] X. Du, M. El-Khamy, and J. Lee. Amnet: Deep atrous multiscale stereo disparity estimation networks. *arXiv preprint arXiv:1904.09099*, 2019. 4322, 4328

[6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 4325

[7] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. *Asian conference on computer vision*, pages 25–38, 2010. 4322, 4324, 4325, 4326

[8] S. Gidaris and N. Komodakis. Detect, replace, refine: Deep structured prediction for pixel wise labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5248–5257, 2017. 4322

[9] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li. Group-wise correlation stereo network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 4322, 4328

[10] R. A. Hamzah and H. Ibrahim. Literature survey on stereo vision disparity map algorithms. *Journal of Sensors*, 2016:1–23, 2016. 4322

[11] D. Hernandez-Juarez, A. Chacón, A. Espinosa, D. Vázquez, J. C. Moure, and A. M. López. Embedded real-time stereo estimation via semi-global matching on the GPU. *International Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA*, pages 143–153, 2016. 4324, 4325, 4326

[12] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007. 4322, 4324, 4325, 4326

[13] 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. *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017. 4322, 4326

[14] S. Khamis, S. Fanello, C. Rhemann, A. Kowdle, J. Valentin, and S. Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 573–590, 2018. 4322, 4328

[15] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 508–515 vol.2, July 2001. 4322

[16] 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. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 4322, 4325, 4328

[17] M. Menze, C. Heipke, and A. Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015. 4325

[18] G.-Y. Nie, M.-M. Cheng, Y. L. Z. Liang, D.-P. Fan, Y. Liu, and Y. Wang. Multi-level context ultra-aggregation for stereo matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 4322, 4328

[19] M. Poggi and S. Mattoccia. Deep stereo fusion: combining multiple disparity hypotheses with deep-learning. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 138–147. IEEE, 2016. 4323

[20] X. Song, X. Zhao, L. Fang, and H. Hu. Edgestereo: A context integrated residual pyramid network for stereo matching. *Asian Conference on Computer Vision*, 2018. 4322, 4328

[21] X. Song, X. Zhao, L. Fang, and H. Hu. Edgestereo: An effective multi-task learning network for stereo matching and edge detection. *arXiv preprint arXiv:1903.01700*, 2019. 4328

[22] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia, and L. Di Stefano. Real-time self-adaptive deep stereo. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4322, 4328

[23] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia. Segstereo: Exploiting semantic information for disparity estimation. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 636–651, 2018. 4322, 4328

[24] Z. Yin, T. Darrell, and F. Yu. Hierarchical discrete distribution decomposition for match density estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 4322, 4328

[25] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 4322, 4323

[26] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, volume 07-12-June-2015, pages 1592–1599. IEEE Computer Society, 10 2015. 4322

[27] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 4328