

Hierarchical Discrete Distribution Decomposition for Match Density Estimation

Zhichao Yin Trevor Darrell Fisher Yu
UC Berkeley

Abstract

Explicit representations of the global match distributions of pixel-wise correspondences between pairs of images are desirable for uncertainty estimation and downstream applications. However, the computation of the match density for each pixel may be prohibitively expensive due to the large number of candidates. In this paper, we propose Hierarchical Discrete Distribution Decomposition (HD³), a framework suitable for learning probabilistic pixel correspondences in both optical flow and stereo matching. We decompose the full match density into multiple scales hierarchically, and estimate the local matching distributions at each scale conditioned on the matching and warping at coarser scales. The local distributions can then be composed together to form the global match density. Despite its simplicity, our probabilistic method achieves state-of-the-art results for both optical flow and stereo matching on established benchmarks. We also find the estimated uncertainty is a good indication of the reliability of the predicted correspondences.

1. Introduction

Finding dense pixel correspondences between two images, typically for stereo matching or optical flow, is one of the earliest problems studied in the computer vision literature. Dense correspondences have wide application including for activity recognition [36], video interpolation [23], scene geometry perception [13], and many others. Challenges when solving this problem include texture ambiguity, complex object motion, illumination change, and entangled occlusion estimation.

Classic approaches jointly optimize local texture matching and neighbor affinity on images [17] possibly in a coarse-to-fine fashion [6, 20, 42]. While these methods can achieve impressive correspondence accuracy, the optimization step may be too slow for downstream applications. Recent works using deep convolutional networks (ConvNets) have achieved similar or even better matching results without an optimization step [10, 21, 40]. Pixel features learned directly from correspondence supervision can capture both local appearance and global context information due to the

large network receptive fields. With GPU acceleration, it is possible to use these networks to regress the pixel displacements in real time [10, 40].

However, the estimation uncertainty inherent in correspondence estimation is neglected by displacement regression approaches. Though post-hoc confidence measures [27, 29] can recover the uncertainty to some degree, they are independent of model training; uncertainty is ignored in the training process. Recognizing the missing uncertainty measures in optical flow methods, some works [12, 43] propose probabilistic frameworks for joint correspondence and uncertainty estimation. Due to constraints on computation and parameter number, they rely on the local Gaussian noise assumption to represent the match distribution. Consequently, they cannot model complicated distributions on a large image area. Early works in stereo matching show that we can build a complete match cost volume as a proxy to estimate the match density, but were not applicable for high-resolution stereo matching nor general optical flow due to the excessive amount of computation needed for the complete cost volume.

In this work, we propose Hierarchical Discrete Distribution Decomposition (HD³), a general probabilistic framework for match density estimation. We aim to find the discrete distribution of possible correspondences with a large support defined on the image grids for each pixel. We adopt a general model to represent pixel-level match probability without any parametric distribution assumption. The model-inherent uncertainty measures can be naturally derived from our estimated match densities.

HD³ decomposes the full match density into multiple levels of local distributions similar to quadtrees. To extract discriminative features for matching, we use networks with Deep Layer Aggregation (DLA) [50] to build the multi-scale feature pyramid. The DLA framework provides us with feature networks of different computation-accuracy trade-offs, which can be easily integrated with other recognition tasks in complex applications. We estimate the match density of the residual motion in each scale, conditioned on match densities at coarser scales. We can propagate the conditional information from previous levels to the prediction at the current level through iterative feature warping and

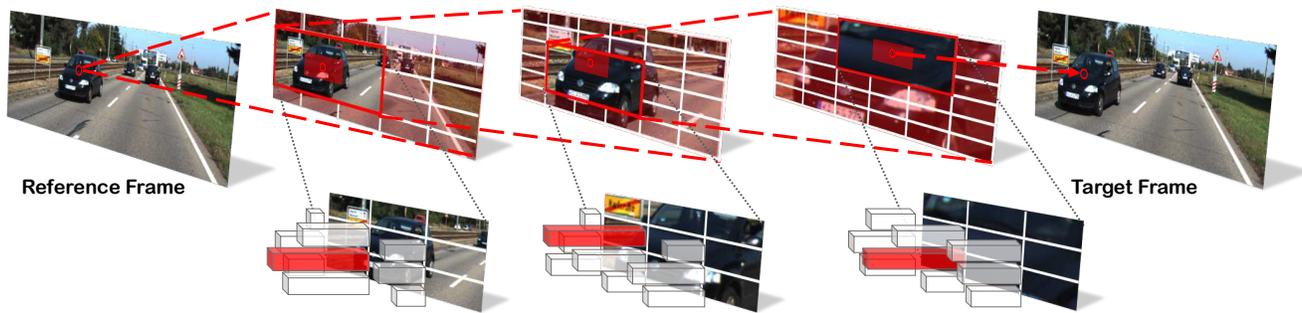


Figure 1: Illustration of HD³. We aim to estimate discrete match distribution in this work. For reducing the infeasible computation cost, the overall distribution is decomposed into multiple scales hierarchically at learning time. The full match information can be recovered by composing predictions from all levels. Please refer to Sec. 3.2 for more details.

density bypass connections. The multi-scale match densities can then be used to recover the complete match density. We can easily convert between point estimates and match densities to train our models on existing datasets with annotations in the form of motion vectors.

We evaluate our framework extensively in two applications: stereo matching and optical flow. Our method achieves state-of-the-art results on both the synthetic dataset MPI Sintel [7] and the real dataset KITTI [13]. Our method not only surpasses all two-frame based optical flow methods by large margins but also beats some competitive scene flow methods on both KITTI 2012 & 2015. We also evaluate our uncertainty estimation and demonstrate the error-awareness of our method in its predictions. Our code is available at <https://github.com/ucbdrive/hd3>.

2. Related Work

Great efforts have been devoted to the problems of finding dense correspondences in the past four decades. For a thorough review, we refer to popular benchmarks including Middlebury [4], MPI Sintel [7], and KITTI [32] benchmarks for both the classical methods and the latest advances in these areas. We will discuss the most related ideas in this section.

Correspondence Estimation. Classical stereo matching usually involves local correspondence extraction and semi-global regularization [16]. On the other hand, optical flow methods typically adopt MRFs [28] to jointly reason about the displacements, occlusions, and symmetries [20, 46] for tackling the more unconstrained and challenging 2D correspondence problem. Despite the distinct differences between the search space dimensions, stereo matching and optical flow share similar assumptions such as brightness constancy and edge-preserving continuity [17, 34, 40].

With the success of deep learning, end-to-end models have been designed for these dense prediction tasks.

Benefiting from pretraining on a large corpus of synthetic data [30], these methods achieve impressive results on par with classical methods [10, 21]. Furthermore, recent advances emphasize the incorporation of classical principles into network designs, such as pyramid matching, feature warping, and contextual regularizer [19, 24, 40]. These improvements contribute to the superior performance of deep learning models, allowing them to surpass classical methods. However, such learning methods neglect the model-inherent uncertainty estimation, *i.e.* they are agnostic of the prediction failure, which is quite important for applications such as autonomous driving and medical imaging. In contrast, our work focuses on the probabilistic correspondence estimation, which can naturally convey the confidence of the predictions.

Uncertainty Measures. Various uncertainty measures have been proposed for classical optical flow estimation. Barron *et al.* [5] proposed a simple method based on the input data characteristics while ignoring the estimated optical flow itself. Kondermann *et al.* [27] learned a probabilistic flow model and obtained uncertainty estimation through hypothesis testing. Mac Aodha *et al.* [29] trained a classifier to assess the prediction quality in terms of end-point-error. These methods either leverage only part of the input information, such as images or predicted flow, for uncertainty estimation, or require post-processing steps independent of model inference itself.

Recently, Gast *et al.* [12] recognized the importance of model-inherent uncertainty measures for deep networks. They proposed probabilistic output layers and employed assumed density filtering to propagate activation uncertainties through the network. For computational tractability, they assumed Gaussian noise and adopted a parametric distribution. Though it can be easily adapted for use with existing regression networks, their performance is only competitive with the deterministic counterparts. Our method provides inherent uncertainty estimation as well as new state-of-the-

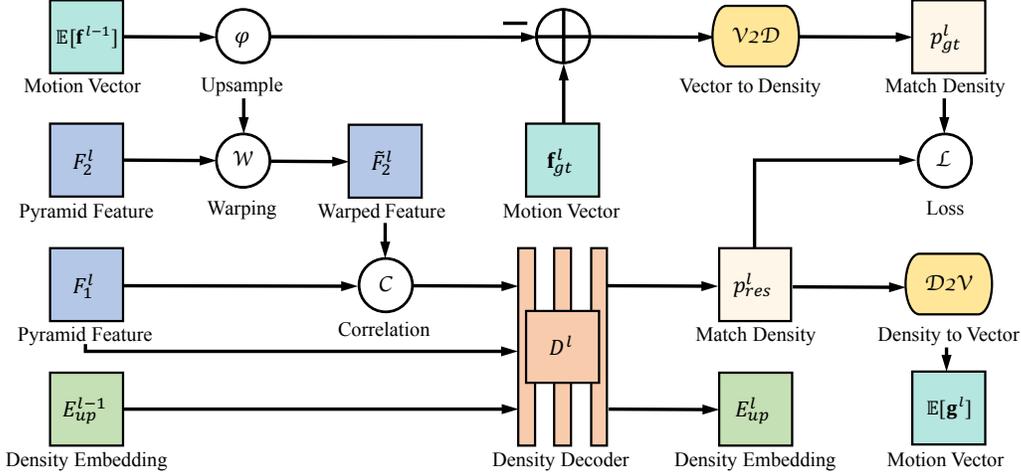


Figure 2: Overview of our architecture. The submodule at the l _{th} level is presented here. F^l and \bar{F}^l denotes the l _{th} level of original and warped pyramid features of image pair I . E_{up}^l denotes upsampled density embeddings between different levels as density bypass connections. f^l and g^l denote motion vectors and p^l corresponds to match density. Their conversion is fulfilled by our $\mathcal{D}2\mathcal{V}$ and $\mathcal{V}2\mathcal{D}$ modules. For details please refer to our method part. This figure is best viewed in color.

art results for both optical flow and stereo matching.

Coarse-to-Fine. Because of the complexity of finding 2D correspondences for each pixel in optical flow, it is natural to match the pixels from coarse to fine resolutions in an image or feature pyramid. This method can be used effectively in the optimization methods [1, 2, 38] as well as patch matching [18]. Its effectiveness is also verified by recent deep learning approaches such as SpyNet [33], PWC-Net [40], and LiteFlowNet [19]. We also estimate our hierarchically decomposed match densities based on the feature pyramid representation [50]. Our contribution lies in the decomposition of the discrete probability distribution instead of the feature representations.

3. Method

In this section, we discuss our probabilistic framework for match density estimation. Without loss of generality, we focus on solving 2D correspondences for optical flow in this section, which can be easily adapted to the 1D case for stereo matching.

3.1. Preliminary

We first introduce the notations and basic concepts used. Given a pair of images $I = \{I_1, I_2\}$, we denote the motion field as $\mathbf{f} = \{\mathbf{f}_{ij}\}$ where $\mathbf{f}_{ij} = (u_{ij}, v_{ij})^T$ for pixels (i, j) , $i = 1, \dots, n, j = 1, \dots, m$. In contrast to Wannenwetsch *et al.* [43] where $\{\mathbf{f}_{ij}\}$ are continuous, we treat them as discrete random variables. We call their density functions *match densities*. We use $p(\mathbf{f}|I)$ to denote the joint probability distribution of $\{\mathbf{f}_{ij}\}$. For brevity, we omit the conditional I in the following discussion when there is no

ambiguity. Finally, we introduce a $\times 2$ upsampling operator φ and an opposite downsampling operator φ^{-1} .

3.2. Match Density Decomposition

The main challenge of estimating the full match density is the prohibitive computational cost. Assume we have an image with size 1000×1000 and displacement range $[-50, 50]$. The cardinality of $\{\mathbf{f}_{ij}\}$ would be 10^6 and the support size of each \mathbf{f}_{ij} could be 10^4 . In this case, the entire distribution volume would have 10 billion cells, which is intractable to generate.

Our key observation is that the full match density can be decomposed hierarchically into multiple levels of distributions. Fig. 1 provides an intuitive illustration. Let us consider multi-scale motion fields $\{\mathbf{f}^l\}$ ($l = 0, \dots, L$), where higher level \mathbf{f}^l has half of the resolution of the lower level \mathbf{f}^{l+1} and \mathbf{f}^L is identical as \mathbf{f} . We introduce a transformation $\mathbf{g}^l = \mathbf{f}^l - \varphi(\mathbf{f}^{l-1})$ to shift the absolute multi-scale motion fields to residual ones. We can recover the original motion field \mathbf{f} from $\{\mathbf{g}^l\}$ ($l = 0, \dots, L$) via

$$\mathbf{f} = \sum_{l=0}^L \varphi^{L-l}(\mathbf{g}^l). \quad (1)$$

Naturally, we have the decomposition of $p(\mathbf{f})$ as

$$p(\mathbf{f}) = \sum_{\{\mathbf{g}^l\} \in \mathbf{F}} \prod_{l=0}^L p(\mathbf{g}^l | \mathbf{G}^{l-1}), \quad (2)$$

where $\mathbf{G}^l = \{\mathbf{g}^s\}_{s=0}^l$, and \mathbf{F} is the set of all possible $\{\mathbf{g}^l\}$ that satisfies Eq. 1. Therefore, we can in turn estimate the decomposed match densities $p(\mathbf{g}^l | \mathbf{G}^{l-1})$ and recover full

match density $p(\mathbf{f})$ through Eq. 2 afterward. The benefit of adopting such decomposition lies in that match density $p(\mathbf{g}^l | \mathbf{G}^{l-1})$ actually has quite low variance, *i.e.* probabilities concentrate on a small subset $R'_{\mathbf{g}^l}$ of the entire support $R_{\mathbf{g}^l}$. Without loss of much information, we can focus on solving $p(\mathbf{g}^l | \mathbf{G}^{l-1})$ with $\mathbf{g}^s \in R'_{\mathbf{g}^s}$ for $s = 0, \dots, l-1$. Consequently, for maximizing the posterior distribution $p(\mathbf{f})$, we achieve satisfactory approximation through maximizing each of the decomposed match densities. We will discuss our selection of support subsets in the next section.

3.3. Learning Decomposed Match Density

Our objective becomes estimating multi-scale decomposed match densities $p(\mathbf{g}^l | \mathbf{G}^{l-1})$. We propose to learn such information through multiple levels of ConvNets. At each level, a ConvNet is designed to estimate the decomposed match density. Note that \mathbf{g}^l is conditioned on \mathbf{G}^{l-1} , while theoretically we can sample $\mathbf{g}^s \in \mathbf{G}^{l-1}$ according to predicted densities at coarser levels.

In this section, we first discuss how to transform point estimate into match density, which is adopted for generating our distribution supervision for each level. Let us consider a *general* motion vector $\mathbf{f}_{ij} \in \mathbf{f}$ and its density function $p(\mathbf{f}_{ij})$. As stated in Sec. 3.2, we prefer $p(\mathbf{f}_{ij})$ to possess a low variance, which would greatly reduce the computation cost through our decomposition. We observe that real-valued \mathbf{f}_{ij} uniquely falls into a 2×2 window W_{ij} in the image grid. This inspires us to splat the bilinear weights of \mathbf{f}_{ij} w.r.t. coordinates in W_{ij} to $p(\mathbf{f}_{ij})$. Concretely, for any $\mathbf{d} \in \mathbb{Z}^2$, we have

$$\mathbb{P}(\mathbf{f}_{ij} = \mathbf{d}) = \begin{cases} 0 & \mathbf{d} \notin W_{ij} \\ \rho(\mathbf{f}_{ij} - \tilde{\mathbf{d}}) & \mathbf{d} \in W_{ij}, \end{cases} \quad (3)$$

where $\rho(\cdot)$ means the product of elements in the vector, and $\tilde{\mathbf{d}}$ is the diagonal opposite coordinate of \mathbf{d} in W_{ij} . We call such conversion as $\mathcal{V}2\mathcal{D}$ (see Fig. 3), which depicts our assumption for the ground-truth match density.

As seen from Eq. 3, the support of $p(\mathbf{f}_{ij})$ is indeed W_{ij} which has a maximum size of 4. Ideally, we can sample $\mathbf{g}^s \in \mathbf{G}^{l-1}$ in a quadtree fashion during estimating the match density of \mathbf{g}^l . However, such computation is still heavy for both training and evaluation. For trade-off, we can discard samplings with minor probabilities. A trivial practice is always taking $\arg \max$ at each level. As a substitution, we propose *local expectation* to further reduce the loss of information. Specifically, for any *general* match density $p(\mathbf{f}_{ij})$, we define W_{ij}^* as the 2×2 window over which the integral of $p(\mathbf{f}_{ij})$ maximizes among all candidate windows. We only retain the probabilities of $p(\mathbf{f}_{ij})$ in W_{ij}^* and normalize it into $p^*(\mathbf{f}_{ij})$. The local expectation is defined as $\mathbb{E}[\mathbf{f}_{ij}]$ w.r.t. $p^*(\mathbf{f}_{ij})$. In the following, we use expectation to denote local expectation by default. We call this conversion

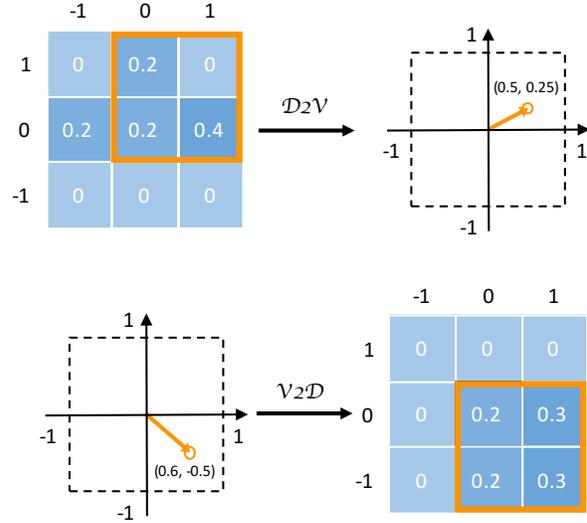


Figure 3: Conversion between motion vectors and match densities. The support is taken as 3×3 here for illustration.

$\mathcal{D}2\mathcal{V}$ (see Fig. 3). Therefore, at each level, instead of exhaustive sampling we always take the max posterior of \mathbf{g}^l as $\mathbb{E}[\mathbf{g}^l]$, and we only estimate $p(\mathbf{g}^l | \mathbf{G}^{(l-1)*})$ (p_{res}^l for short in the following) in each level, where $\mathbf{G}^{(l-1)*} = \{\mathbb{E}[\mathbf{g}^s]\}_{s=0}^{l-1}$. This enables us to get rid of expensive training and test time sampling.

3.4. Network Architecture Design

Finally, we discuss the network architecture design for estimating the decomposed match densities. We achieve this objective via stacking multiple levels of ConvNets, which we call density decoders $\{D^l\}$. D^l infers the match density p_{res}^l in its respective level l . A single level of the entire network is illustrated in Fig. 2. In the following, we discuss the details of our subnetworks.

The architecture design of our density decoder D^l is motivated by the close relationship between the targeted output p_{res}^l and the similarity information between image pairs, or their embedded representations. Our match density estimation operates on multi-scale feature embeddings $\{F_1^l, F_2^l\}$, which are extracted via a DLA [50] network over $\{I_1, I_2\}$. Affinity information can be obtained through the correlation [10] of feature embeddings between different frames. For performing long-range correlation and imposing conditional priors from previous levels, we always warp the feature F_2^l according to $\varphi(\mathbb{E}[\mathbf{f}^{l-1}])$ before correlation. The cost volume is concatenated with F_1^l , $\varphi(\mathbb{E}[\mathbf{f}^{l-1}])$, and the upsampled *density embedding* E_{up}^{l-1} from the previous level, then fed into our density decoder D^l . The decoder D^l produces the *density embedding*, from which we obtain the match density via a classifier. Also, we upsample the *den-*

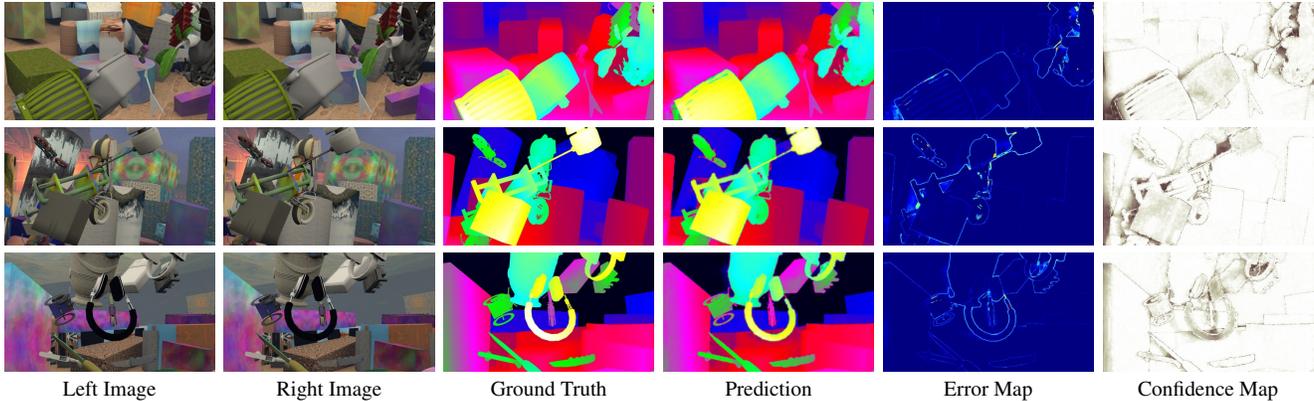


Figure 4: Visualized stereo results on the validation set of FlyingThings3D. Cold colors in the error map denote correct predictions while warm colors mean the contrary. Our network gives accurate results in most regions, while errors tend to occur at boundaries and occlusions.

city embedding to E_{up}^l and feed it to the next level as density bypass connections. Both of the pyramid feature extractor and the density decoder are jointly trained in an end-to-end manner.

At inference time, we calculate $\mathbb{E}[\mathbf{g}^l]$ from each predicted p_{res}^l and compose them via Eq. 1 to produce the point estimate of \mathbf{f} . While during training time, we downsample the ground-truth motion field into \mathbf{f}_{gt}^l . The residual motion w.r.t. $\varphi(\mathbb{E}[\mathbf{f}^{l-1}])$ is converted into p_{gt}^l . The entire training loss comes in the form of Kullback–Leibler divergence

$$\mathcal{L} = \sum_l \sum_{\mathbf{g} \in R_{\mathbf{g}^l}} p_{\text{gt}}^l(\mathbf{g}) (\log p_{\text{gt}}^l(\mathbf{g}) - \log p_{\text{res}}^l(\mathbf{g})). \quad (4)$$

4. Experiments

HD³ provides hierarchically decomposed match densities. It can be used for different tasks, such as stereo matching and optical flow. The probability of point estimates can be used as uncertainty estimation. It is hard to evaluate the quality of the learned distribution directly, but we can investigate its performance being applied to these specific tasks.

4.1. Implementation Details

Network Variants. We can apply our models to stereo matching and optical flow. The networks are called HD³S and HD³F. The two variants differ slightly: we adopt 1D correlation for HD³S and 2D correlation for HD³F. The correlation range is always 4 for both tasks at different levels, which is consistent with the size of match density support. Since we treat stereo matching as 1D flow estimation, we clip the positive values in converted point estimates at each level for HD³S. The pyramid level is set to 5 for HD³F and 6 for HD³S based on experiment results.

Module Details. We select DLA-34-Up [50] as our pyramid feature extractor, because it can achieve competitive se-

mantic segmentation accuracy on small datasets with much less computation than the deeper alternatives. The features at the coarsest level are $\times 64$ downsampled. The density decoder D^l consists of two residual blocks plus one aggregation node [15, 50], except for the last level when it is fulfilled via a dilated convolutional network [49] as a context module. We adopt batch normalization [22] in all of our models to stabilize the training. Predictions are upsampled from the lowest level with highest output resolution to full resolution during evaluation.

Training Details. We train our models on 8 GPUs without synchronized batch normalization. The weights of pyramid feature extractor are initialized from the ImageNet pre-trained model. The network is optimized by Adam [25], where $\beta_1 = 0.9$, $\beta_2 = 0.999$. For all of our pretraining experiments on synthetic datasets, models are trained for 200 epochs, and the learning rate is decayed by 0.5 every 30 epochs after 70 epochs for 4 times in total. As for data augmentation, besides random cropping, we adopt random resizing and color perturbation [31] during the fine-tuning stage, and introduce random flipping for optical flow experiments. The dense and sparse annotations, as supervision at different scales, are bilinearly downsampled and average pooled from the ground-truth map respectively. In this section, unless otherwise stated, confidence maps are obtained through aggregating the probabilities within W_i^* of the last level match density, and uncertainty maps are the opposite.

4.2. Stereo Matching

To evaluate the performance of our HD³S model, we benchmark our result on the KITTI stereo dataset [13]. Due to the limited amount of training data in KITTI, we pretrain our model on the FlyingThings3D dataset [30].

FlyingThings3D. We use the FlyingThings3D dataset as

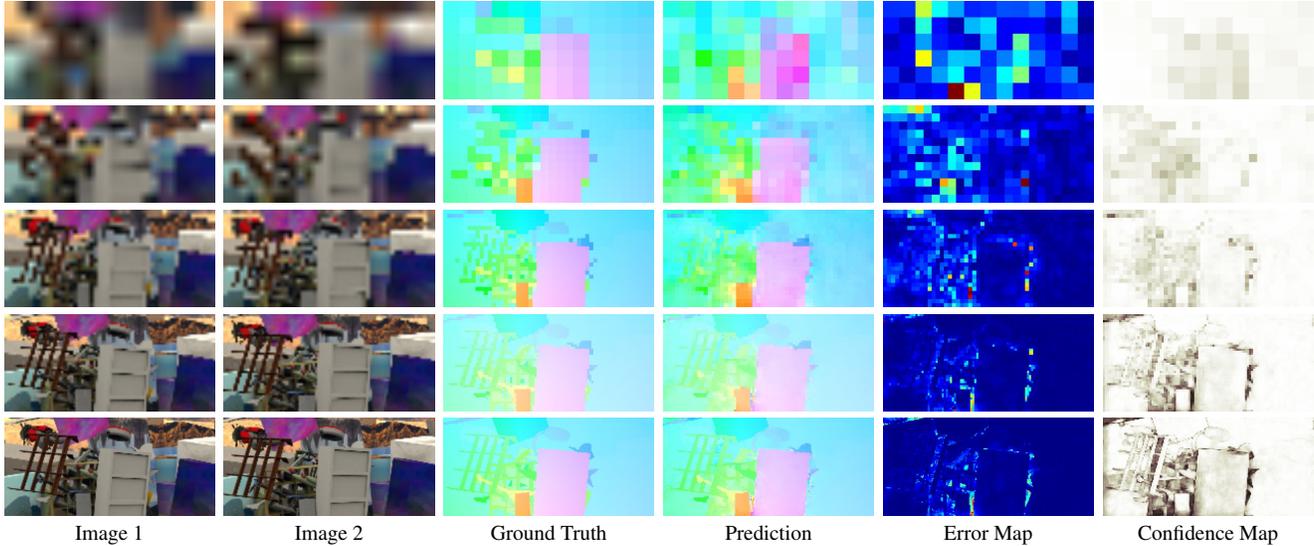


Figure 5: Qualitative multi-scale flow result on the validation set of FlyingThings3D dataset. Bilinearly downsampled raw images, coarser level flows, error maps and confidence maps are enlarged via nearest neighbor upsampling for visualization purpose. Our network gives precise predictions in most regions, while occasionally presents confusion in occluded regions and disappearing parts.

	KITTI 2012		KITTI 2015			Time
Methods	Out-Noc	Out-All	D1-bg	D1-fg	D1-all	(s)
SPS-st [46]	3.39	4.41	3.84	12.67	5.31	2.00
Displets v2 [14]	2.37	3.09	3.00	5.56	3.43	265
MC-CNN-acrt [51]	2.43	3.63	2.89	8.88	3.88	67.0
SGM-Net [35]	2.29	3.50	2.66	8.64	3.66	67.0
L-ResMatch [37]	2.27	3.40	2.72	6.95	3.42	48.0
GC-Net [24]	1.77	2.30	2.21	6.16	2.87	0.90
EdgeStereo [39]	1.73	2.18	2.27	4.18	2.59	0.27
PDSNet [41]	1.92	2.53	2.29	4.05	2.58	0.50
PSMNet [8]	1.49	1.89	1.86	4.62	2.32	0.41
SegStereo [47]	1.68	2.03	1.88	4.07	2.25	0.60
HD³S (Ours)	1.40	1.80	1.70	3.63	2.02	0.14

Table 1: Stereo matching results on KITTI test set. All of the numbers denote percentages of disparity outliers. The official leaderboard ranks methods according to “Out-Noc” for KITTI 2012 and “D1-all” for KITTI 2015.

training data. Following the training protocol of the original FlowNet2 model [21], we use a subset of the dataset which omits some extremely hard samples. We train our model with a batch size of 32 and an initial learning rate of 2×10^{-4} . The image crop size is 320×896 . Qualitative examples, as well as the confidence maps, are shown in Fig. 4. We find low confidence correlates well with prediction errors, which generally occurs at boundaries and occlusions.

KITTI. During fine-tuning stage, we leverage the available 394 image pairs from KITTI 2012 & 2015 as training data. Training is performed for 2000 epochs, with batch size 16

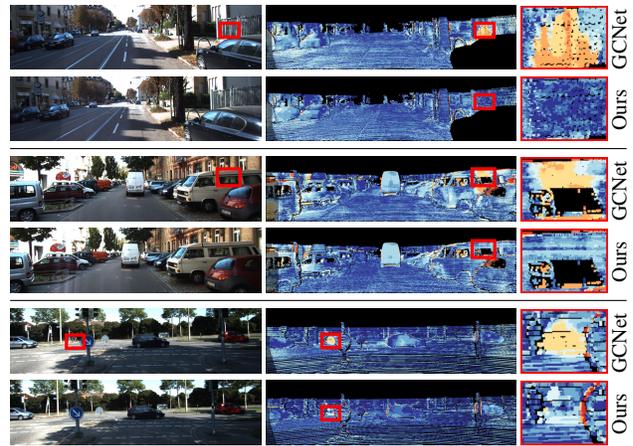


Figure 6: Example stereo error maps on KITTI 2015 test set. We contrast our method with GC-Net [24]. Orange corresponds to erroneous prediction. This figure is best viewed in color.

and image crop size 320×896 . The initial learning rate is 1×10^{-5} and decayed by 0.5 at the 1000th and the 1500th epoch.

As shown in Tab. 1, our method achieves the lowest percentages of disparity outliers in both non-occluded (Out-Noc) and total regions (Out-All), background (D1-bg) and foreground regions (D1-fg), among all of the competitive baselines on both datasets. We also hold the lowest inference time for processing a standard KITTI stereo pair. Note that we do not leverage the entire Scene Flow dataset [30] for training as [8, 24], nor do we utilize additional semantic or edge cues as in [39, 47]. Qualitative comparisons are

Methods	Training		Test		Time (s)
	Clean	Final	Clean	Final	
PatchBatch [11]	-	-	5.79	6.78	50.0
EpicFlow [34]	-	-	4.12	6.29	15.0
CPM-flow [18]	-	-	3.56	5.96	4.30
FullFlow [9]	-	3.60	2.71	5.90	240
FlowFields [2]	-	-	3.75	5.81	28.0
MRFflow [44]	1.83	3.59	2.53	5.38	480
FlowFieldsCNN [3]	-	-	3.78	5.36	23.0
DCFlow [45]	-	-	3.54	5.12	8.60
SpyNet-ft [33]	(3.17)	(4.32)	6.64	8.36	0.16
FlowNet2 [21]	2.02	3.14	3.96	6.02	0.12
FlowNet2-ft [21]	(1.45)	(2.01)	4.16	5.74	0.12
LiteFlowNet [19]	2.52	4.05	-	-	0.09
LiteFlowNet-ft [19]	(1.64)	(2.23)	4.86	6.09	0.09
PWC-Net [40]	2.55	3.93	-	-	0.03
PWC-Net-ft [40]	(2.02)	(2.08)	4.39	5.04	0.03
HD ³ F (Ours)	3.84	8.77	-	-	0.08
HD ³ F-ft (Ours)	(1.70)	(1.17)	4.79	4.67	0.08

Table 2: Average EPE results on MPI Sintel dataset. “-ft” means finetuning on the Sintel training set and numbers in the parenthesis are results on data the method has been trained on.

shown in Fig. 6. Our method exhibits better performance in regions with complex and ambiguous textures. This indicates the effectiveness of hierarchical match density learning based on pyramid feature representations, which exhibits robustness to local noise.

4.3. Optical Flow

We pretrain our HD³F on synthetic data from FlyingChairs [10] and FlyingThings3D [21], then investigate the effectiveness of our model on established optical flow benchmarks including MPI Sintel [7] and KITTI [13].

FlyingChairs. We train our network on FlyingChairs with batch size 64 and initial learning rate 4×10^{-4} . Images are randomly resized and cropped to 384×512 patches. We find larger crop size can improve the network performance.

FlyingThings3D. We further fine-tune the model on the FlyingThings3D data, the same subset in our stereo matching experiments, with batch size 32, learning rate 4×10^{-5} and image crop size 384×832 . We visualize examples of multi-scale predictions in Fig. 5. The results indicate that our model is able to progressively refine the prediction from coarse to fine scales. Though we adopt the discrete distribution, our model can still capture very detailed displacements.

MPI Sintel. Finally, we fine-tune our model on MPI Sintel [7] for 1200 epochs with batch size 32 and image crop size 384×768 . The initial learning rate is 2×10^{-5} and

Methods	KITTI 2012			KITTI 2015		
	AEPE	AEPE	F1-Noc	AEPE	F1-all	F1-all
	<i>train</i>	<i>test</i>	<i>test</i>	<i>train</i>	<i>train</i>	<i>test</i>
EpicFlow [34]	-	3.8	7.88%	-	-	26.29%
FullFlow [9]	-	-	-	-	-	23.37%
PatchBatch [11]	-	3.3	5.29%	-	-	21.07%
FlowFields [2]	-	-	-	-	-	19.80%
DCFlow [45]	-	-	-	-	15.09%	14.83%
MirrorFlow [20]	-	2.6	4.38%	-	9.93%	10.29%
PRSM [42]	-	1.0	2.46%	-	-	6.68%
SpyNet-ft [33]	(4.13)	4.7	12.31%	-	-	35.07%
FlowNet2 [21]	4.09	-	-	10.06	30.37%	-
FlowNet2-ft [21]	(1.28)	1.8	4.82%	(2.30)	(8.61%)	10.41%
LiteFlowNet [19]	4.25	-	-	10.46	29.30%	-
LiteFlowNet [19]	(1.26)	1.7	-	(2.16)	(8.16%)	10.24%
PWC-Net [40]	4.14	-	-	10.35	33.67%	-
PWC-Net-ft [40]	(1.45)	1.7	4.22%	(2.16)	(9.80%)	9.60%
HD ³ F (Ours)	4.65	-	-	13.17	23.99%	-
HD ³ F-ft (Ours)	(0.81)	1.4	2.26%	(1.31)	(4.10%)	6.55%

Table 3: Optical flow results on KITTI dataset. “-ft” means finetuning on the KITTI training set. Numbers in parenthesis are results on data the network has been trained on.

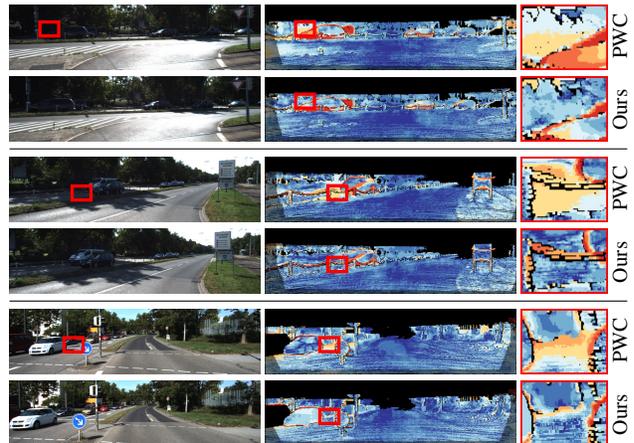


Figure 7: Example flow error maps on KITTI 2015 test set. We compare our method with PWC-Net [40]. Orange corresponds to erroneous prediction. This figure is best viewed in color.

decayed by 0.5 at the 600th and the 900th epoch. Though the dataset provides training data of different subsets (*clean* & *final* passes), we only adopt the *final* pass as training data rendered with motion blur, defocus blur, and atmospheric effects. As shown in Tab. 2, we can obtain the lowest average EPE in the *final* pass, and compelling results on the *clean* pass, though our model does not see the *clean* pass data during training. In the model generalization experiment, our pretrained HD³F estimates the flow accurately near the occlusion boundary, resulting in the lowest out-

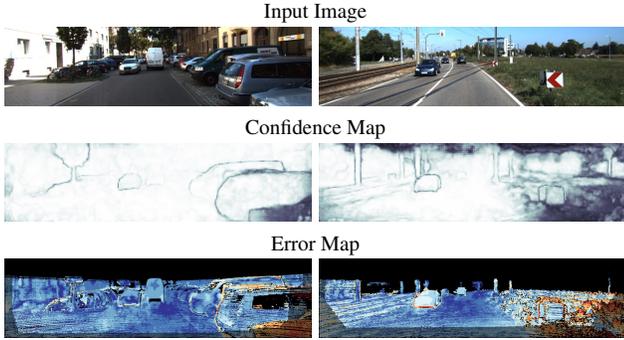


Figure 8: Example confidence maps of our predictions and error maps w.r.t. ground-truth. In confidence maps, white colors mean confident predictions while dark colors denote uncertain ones. In the error maps, warmer colors indicate inaccurate predictions.

lier percentage on KITTI (see the “HD³F (Ours)” entry in Tab. 3). The metric of EPE emphasizes large motion error. This influence makes our pretrained HD³F achieve higher EPE on MPI Sintel (see the “HD³F (Ours)” entry in Tab. 2).

KITTI. Alternatively, we can finetune our pretrained model on KITTI dataset. We follow the configurations of our stereo experiment. Tab. 3 summarize the results. We can obtain the lowest F1-Noc on KITTI 2012 test set and the lowest F1-all on KITTI 2015 test set. At the time of writing, HD³F outperforms all two-frame optical flow methods by large margins on both KITTI 2012 & 2015. It even surpasses some competitive scene flow methods such as PRSM [42], which use additional stereo data. This reveals the suitability of our probabilistic method in challenging real-world cases. We show qualitative comparisons against PWC-Net in Fig. 7. Our method appear to have advantages in estimating many thin structures.

4.4. Uncertainty Estimation

We also conduct quantitative analysis of uncertainty estimation. We compute the log likelihoods of our network predictions and compare HD³F with probabilistic flow networks [12]. FlowNetDropOut uses variational Gaussian dropout layers [26]. While FlowNetProbOut replaces deterministic outputs with probabilistic output layers. FlowNetADF propagates uncertainty through the entire network using ADF. During the evaluation, we recover the full match density through composing the multi-scale match densities. This can be achieved through iteratively sampling from coarse to fine, and we assume a discrete non-uniform distribution for sampling outside W_i^* (see Sec. 3.3). As shown in Tab. 4, HD³F achieves the best average log likelihoods against all of the baselines.

Furthermore, we measure the reliability of network prediction based on uncertainty. We treat predictions with uncertainty greater than a certain threshold ($\sigma = 0.3$) as out-

Methods	Sintel clean	Sintel final	Chairs
FlowNetDropOut [12]	-7.106	-10.820	-6.176
FlowNetProbOut [12]	-6.888	-7.621	-3.591
FlowNetADF [12]	-3.878	-4.186	-3.348
HD³F(Ours)	-1.487	-1.821	-0.872

Table 4: Average log likelihoods of probabilistic flow methods on MPI Sintel training set and FlyingChairs test set.

Classes	Methods	Noc		All	
		IoU	Acc	IoU	Acc
Outlier	Consistency	17.5	64.9	23.3	81.9
	Ours	37.6	57.8	44.1	76.4
Inlier	Consistency	84.2	85.8	75.6	76.9
	Ours	96.1	97.8	91.8	93.7
Mean	Consistency	50.9	75.4	49.5	79.4
	Ours	66.9	77.8	68.0	85.1

Table 5: Classification result of inlier and outlier predictions on KITTI 2015 training set. Noc denotes evaluation only in the non-occluded area, while All denotes evaluation in the overall region.

liers and compare such criterion with the forward-backward consistency check [48] which is popularly adopted for point estimate. Both methods use the same HD³F model for inference. As shown in Tab. 5, our uncertainty estimation gives the highest mean IoU and mean accuracy in both non-occluded and overall regions. Fig. 8 presents visualization of the confidence and error maps. We can observe the positive correlation between our estimated uncertainty and prediction error.

5. Conclusion

We proposed Hierarchical Discrete Distribution Decomposition (HD³) for estimating the match density. Our approach decomposed the match density into multiple scales and learned the decomposed match densities in an end-to-end manner. The predicted match densities can be converted into point estimates, while providing model-inherent uncertainty measures at the same time. Our experiments demonstrated the advantages of our method on established benchmarks.

In the future, we hope to integrate more information into our framework such as the pixel assignment probabilities from segmentation. Currently, we do not consider relationships between match densities of adjacent pixels, but this may help remove match uncertainty in challenging cases.

Acknowledgments This work was supported by Berkeley AI Research and Berkeley DeepDrive.

References

- [1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *IJCV*, 1989. 3
- [2] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *ICCV*, 2015. 3, 7
- [3] C. Bailer, K. Varanasi, and D. Stricker. Cnn-based patch matching for optical flow with thresholded hinge embedding loss. In *CVPR*, 2017. 7
- [4] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 2011. 2
- [5] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *IJCV*, 1994. 2
- [6] A. Behl, O. Hosseini Jafari, S. Karthik Mustikovela, H. Abu Alhaija, C. Rother, and A. Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In *ICCV*, 2017. 1
- [7] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 2, 7
- [8] J.-R. Chang and Y.-S. Chen. Pyramid stereo matching network. In *CVPR*, 2018. 6
- [9] Q. Chen and V. Koltun. Full flow: Optical flow estimation by global optimization over regular grids. In *CVPR*, 2016. 7
- [10] 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. 1, 2, 4, 7
- [11] D. Gadot and L. Wolf. Patchbatch: a batch augmented loss for optical flow. In *CVPR*, 2016. 7
- [12] J. Gast and S. Roth. Lightweight probabilistic deep networks. In *CVPR*, 2018. 1, 2, 8
- [13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 1, 2, 5, 7
- [14] F. Güney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *CVPR*, 2015. 6
- [15] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 5
- [16] H. Hirschmüller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *PAMI*, 2009. 2
- [17] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 1981. 1, 2
- [18] Y. Hu, R. Song, and Y. Li. Efficient coarse-to-fine patch-match for large displacement optical flow. In *CVPR*, 2016. 3, 7
- [19] T.-W. Hui, X. Tang, and C. C. Loy. LiteFlowNet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, 2018. 2, 3, 7
- [20] J. Hur and S. Roth. Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. In *ICCV*, 2017. 1, 2, 7
- [21] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 1, 2, 6, 7
- [22] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5
- [23] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz. Super slo-mo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, 2018. 1
- [24] 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. 2, 6
- [25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, 2014. 5
- [26] D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *NIPS*, 2015. 8
- [27] C. Kondermann, R. Mester, and C. Garbe. A statistical confidence measure for optical flows. In *ECCV*, 2008. 1, 2
- [28] S. Z. Li. Markov random field models in computer vision. In *ECCV*, 1994. 2
- [29] O. Mac Aodha, A. Humayun, M. Pollefeys, and G. J. Brostow. Learning a confidence measure for optical flow. *PAMI*, 2013. 1, 2
- [30] 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. 2, 5, 6
- [31] S. Meister, J. Hur, and S. Roth. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, 2018. 5
- [32] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 2
- [33] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017. 3, 7
- [34] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015. 2, 7
- [35] A. Seki and M. Pollefeys. Sgm-nets: Semi-global matching with neural networks. In *CVPR*, 2017. 6
- [36] L. Sevilla-Lara, Y. Liao, F. Güney, V. Jampani, A. Geiger, and M. J. Black. On the integration of optical flow and action recognition. In *GCPR*, 2018. 1
- [37] A. Shaked and L. Wolf. Improved stereo matching with constant highway networks and reflective confidence learning. In *CVPR*, 2017. 6
- [38] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger. Probability distributions of optical flow. In *CVPR*, 1991. 3
- [39] X. Song, X. Zhao, H. Hu, and L. Fang. Edgestereo: A context integrated residual pyramid network for stereo matching. In *ACCV*, 2018. 6
- [40] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 1, 2, 3, 7
- [41] S. Tulyakov, A. Ivanov, and F. Fleuret. Practical deep stereo (pds): Toward applications-friendly deep stereo matching. In *NeurIPS*, 2018. 6

- [42] C. Vogel, K. Schindler, and S. Roth. 3d scene flow estimation with a piecewise rigid scene model. *IJCV*, 2015. 1, 7, 8
- [43] A. S. Wannenwetsch, M. Keuper, and S. Roth. Probflow: Joint optical flow and uncertainty estimation. In *ICCV*, 2017. 1, 3
- [44] J. Wulff, L. Sevilla-Lara, and M. J. Black. Optical flow in mostly rigid scenes. In *CVPR*, 2017. 7
- [45] J. Xu, R. Ranftl, and V. Koltun. Accurate optical flow via direct cost volume processing. *CoRR*, 2017. 7
- [46] K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *ECCV*, 2014. 2, 6
- [47] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia. Segstereo: Exploiting semantic information for disparity estimation. In *ECCV*, 2018. 6
- [48] Z. Yin and J. Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *CVPR*, 2018. 8
- [49] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2015. 5
- [50] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep layer aggregation. In *CVPR*, 2018. 1, 3, 4, 5
- [51] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *JMLR*, 2016. 6