Seminar on Current Works in Computer Vision

Hierarchical Discrete Distribution Decomposition for Match Density Estimation

Albert-Ludwigs-Universität Freiburg

Presenter: Saurav Shanu

Authors

Zhichao Yin

Trevor Darrell

Fisher Yu

UNI FREIBURG

Table of contents

Introduction

- Find dense correspondences for
  - Stereo Matching
  - Optical Flow

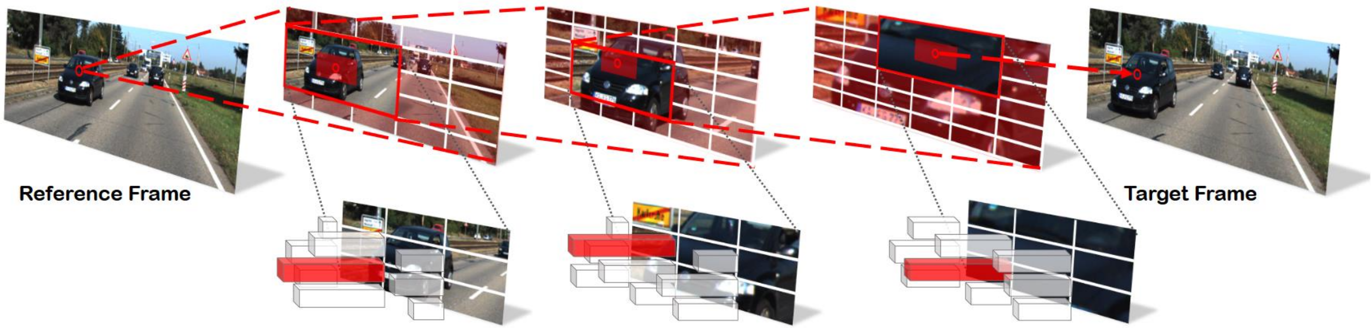
  with match density estimation

Introduction -- HD^3

- Method is "*Hierarchical Discrete Distribution Decomposition.*"
- A general probabilistic frame-work for match density estimation.
- Model-inherent match density estimation.

HD^3

- Hierarchical decomposition of the image.
- Calculate match density at each level.
- Calculate full match density.



Why decomposing Match Density?

- Estimating full match density can be computationally expensive.
- For a 1000 × 1000 image,
  - With displacement range of motion vector $f_{ij,}$ [-50, 50]
  - Total number of cells to compute, 10^{10}.

Table of contents

- Introduction
- Related Work
- Method
- Experiments
- Summary

Related Work

- Effective deep learning models for flow estimation are available.
- Some of them do uncertainty estimation as well.
- They lack non-parametric match density.

Ilg et al. 2016

Early work on uncertainty estimation

- Need for confidences established by Barron et al. IJCV 1994.
- Confidence measure was mostly based on the input image like
  - Gradient of the image sequence
  - Hessian of the image sequence
  - the trace or smallest eigenvalue of the structure tensor.
- For example, in the Lucas-Kanade method confidence is based on the gradient of the image.

Other methods for uncertainty estimation

- Kondermann et al. (ECCV, 2008) gave uncertainty estimation through hypothesis testing.
  - Hypothesis - “The central flow vector of a given flow field patch follows the underlying conditional distribution given the remaining flow vectors of the patch.”

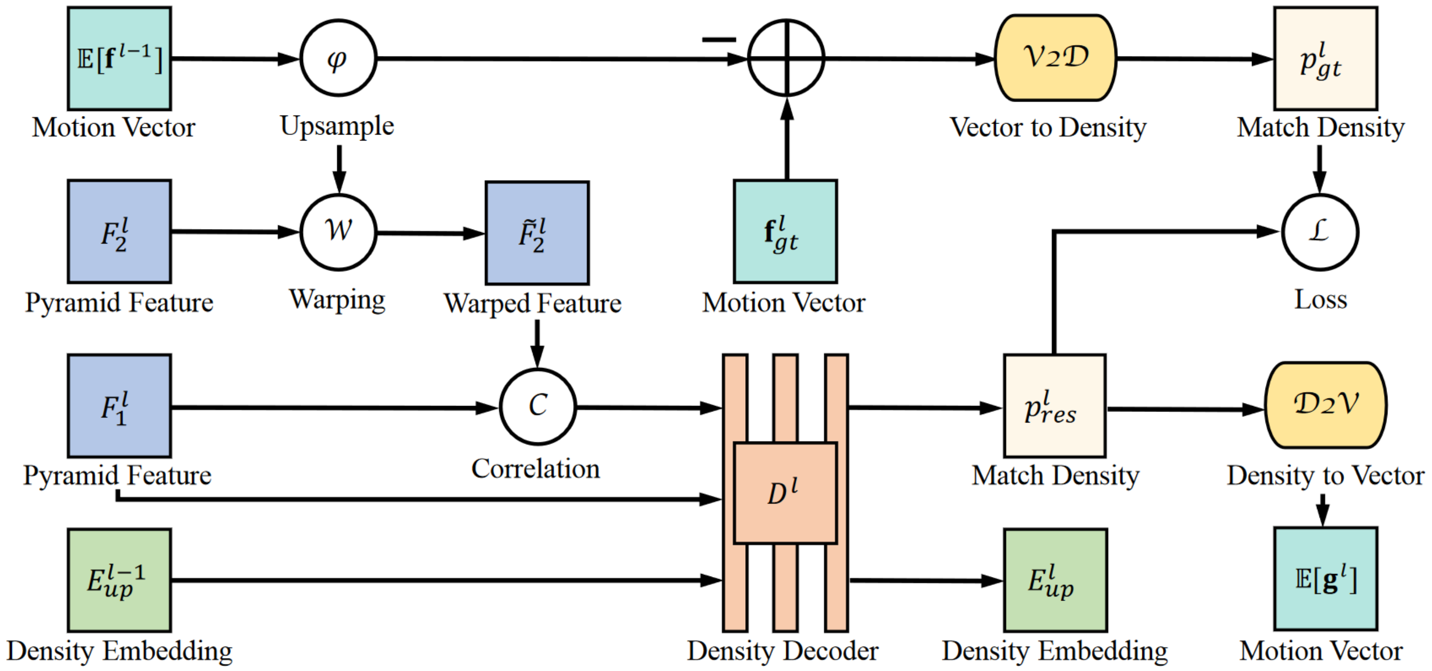
- Mac Aodha et al. trained a classifier to assess the prediction quality in terms of end-point-error.
  - Confidence for each flow vector as the probability of that flow being below some specified error threshold.

Model inherent uncertainty measure

- Gast et al. CVPR 2018 proposed probabilistic output layers.
- For computational tractability, they assumed Gaussian noise and adopted a parametric distribution.
- Performance is only competitive with the deterministic counterparts.

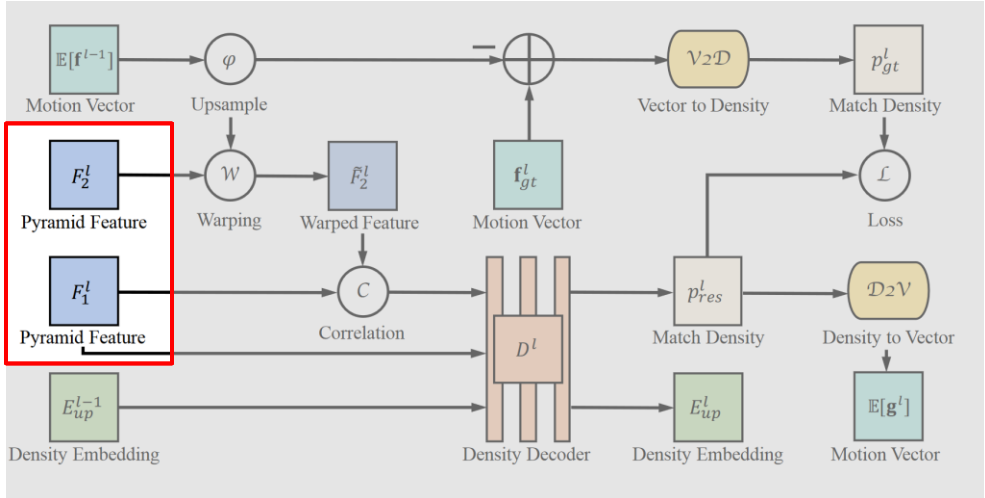
Table of contents

Network Architecture – Single layer

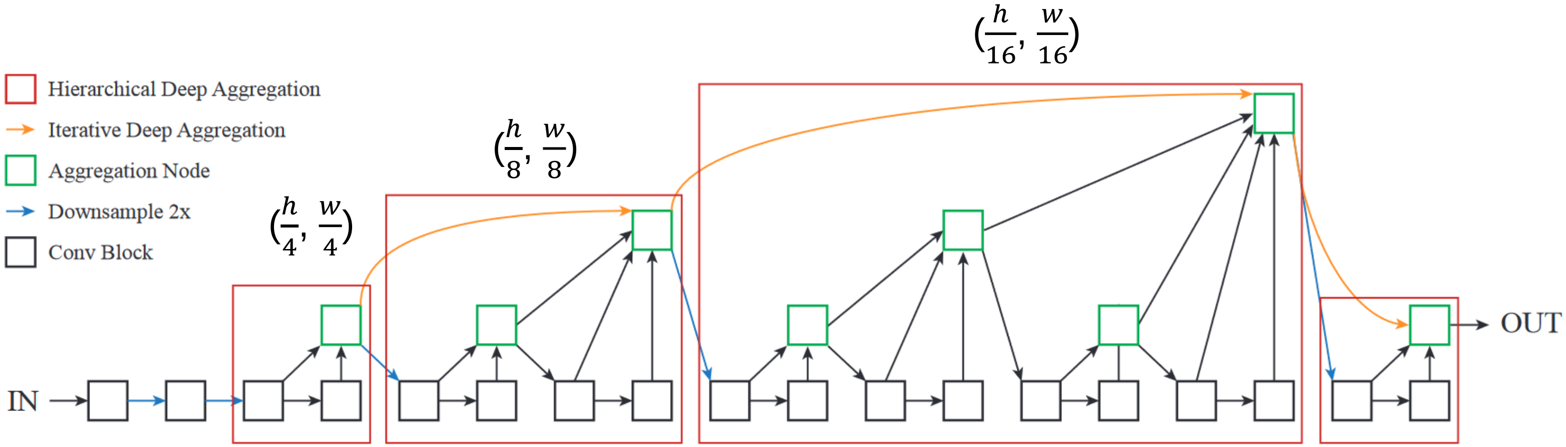


Extract multi-scale features from images

- Extract features F_1 and F_2 from I_1 and I_2 via DLA (Fisher et al. CVPR 2018)

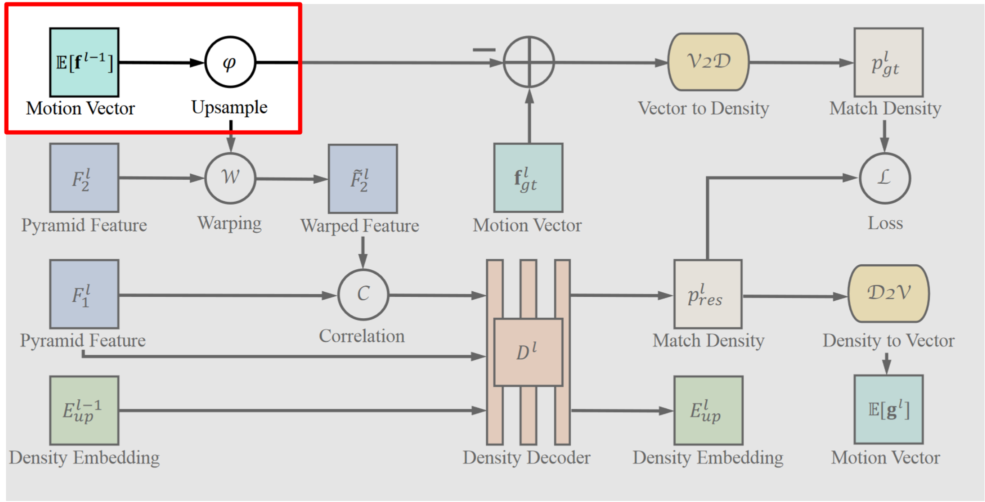


DLA – Deep layer aggregation



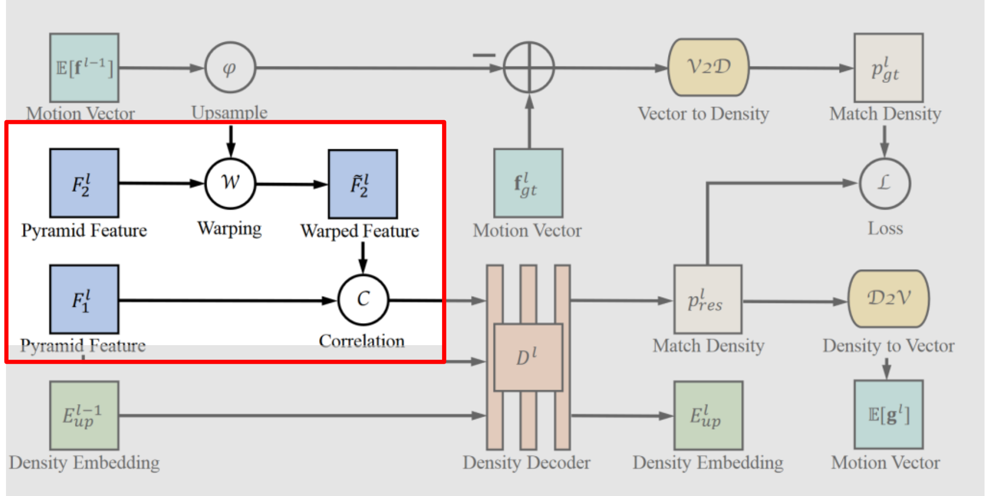
Upscale motion vector

- Get motion vector from previous layer.
- Upsample it to current layer size.



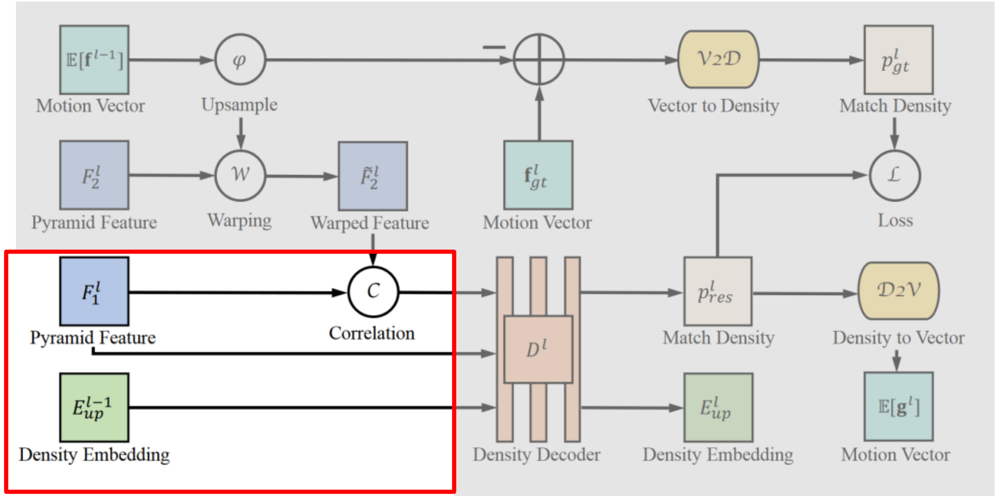
Find correlation

- Find correlations between
  - reference frame features.
  - warped target frame features.



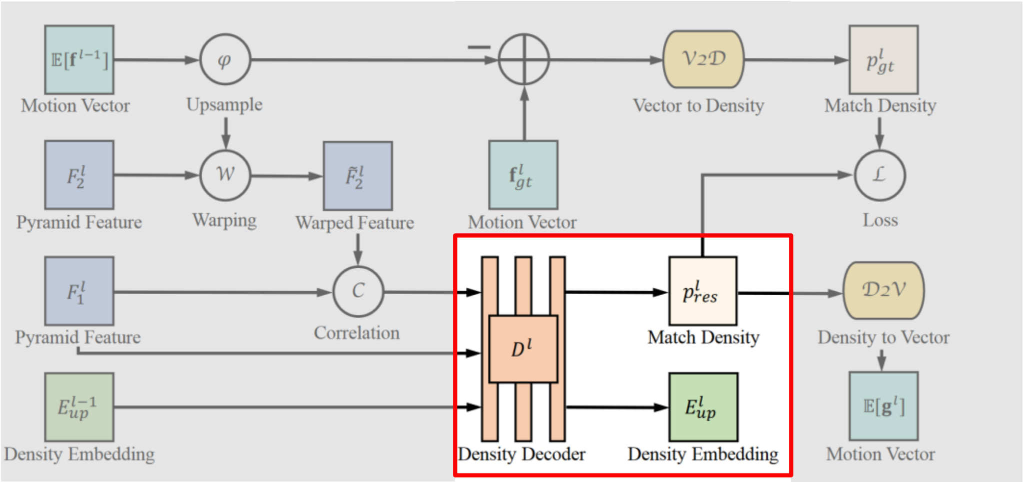
Feed to Density Decoder

- Feed cost volume, features F_1 and density embedding to the density decoder.
- Use density embedding from previous layer.



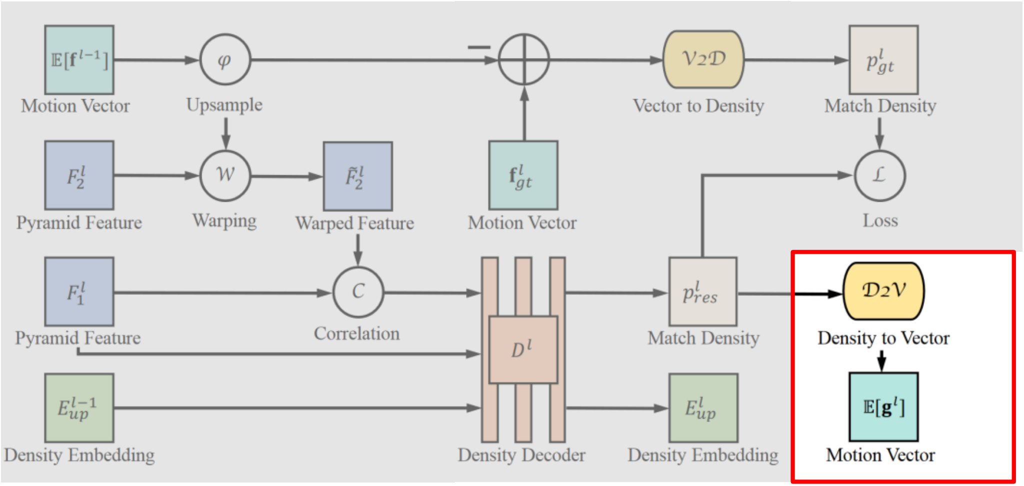
Density decoder

- Density decoder is a ConvNet which gives density embedding and match density.



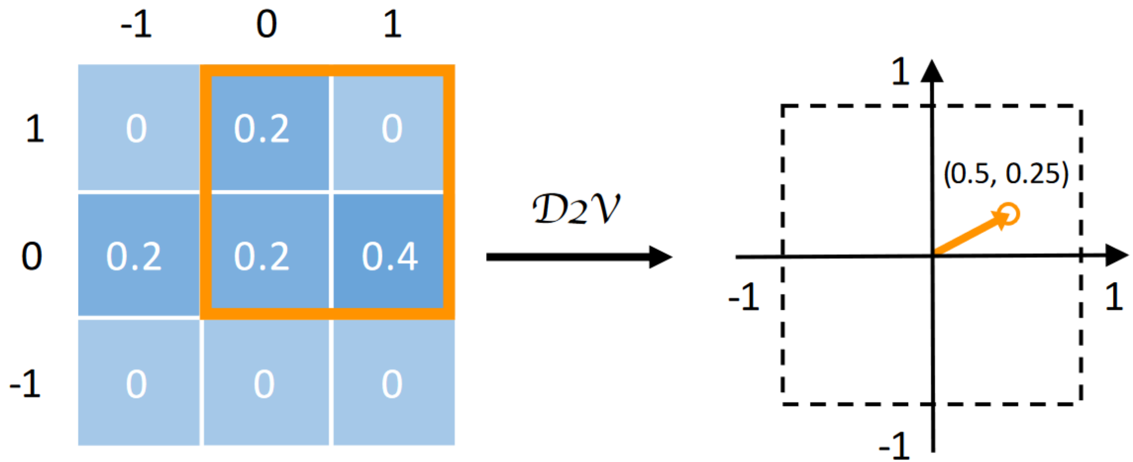
Retrieving Motion vector

- Motion vector is calculated from the D2V method using match densities.



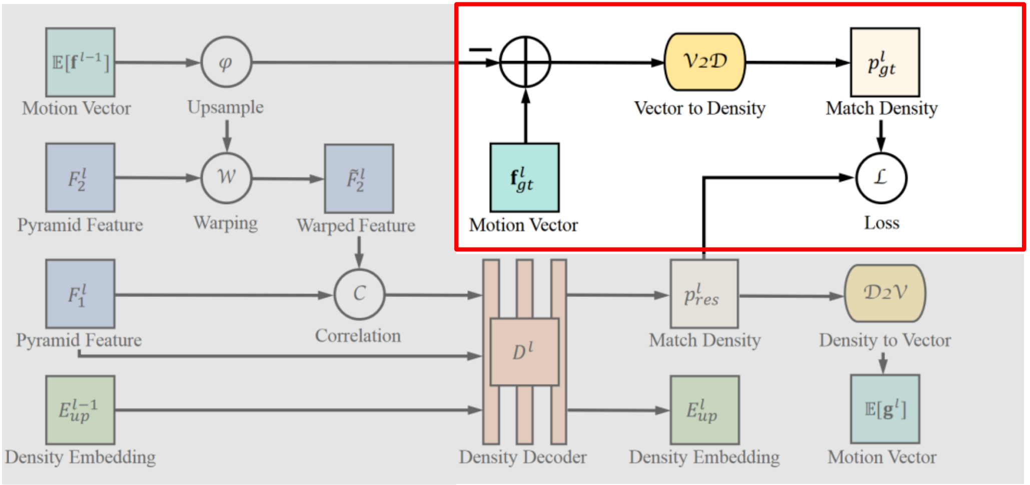
D2V- Density to Vector

- Converts match density to vector.



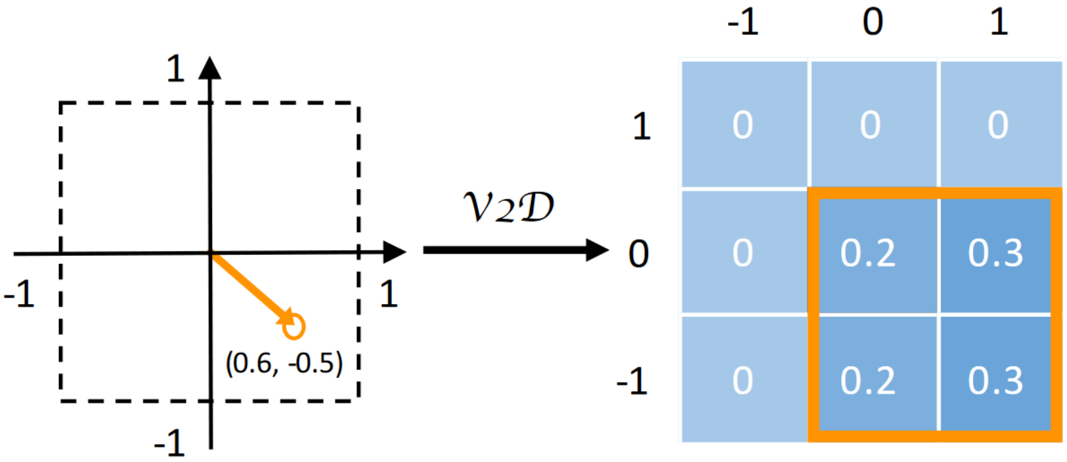
Calculate Loss

- Ground truth motion vector is converted to match density by V2D method.
- Loss between both densities is then calculated.



V2D – Vector to Density

- Converts ground truth motion vector to ground truth match density.



Loss

- Loss is represented as Kullback-Leibler divergence.

$$\mathcal{L} = \sum_{l} \sum_{\mathbf{g} \in R_{\mathbf{g}^l}} p^l_{\texttt{gt}}(\mathbf{g})(\log p^l_{\texttt{gt}}(\mathbf{g}) - \log p^l_{\texttt{res}}(\mathbf{g})).$$

where,

$$\mathbf{g}^l = \mathbf{f}^l - \varphi(\mathbf{f}^{l-1})$$

$\mathbf{f}^l$	flow vector at layer *l.*
φ	upsampling operator.
l	layers in network.
$R_{\mathbf{g}^l}$	Support set of g

Table of contents

- Introduction
- Related Work
- Method
- Experiments
- Summary

General setup

- Entire network is trained in an end-to-end manner.
- The weights of pyramid feature extractor are initialized from the ImageNet pre-trained model.
- Same scheme is applied to both stereo matching and optical flow with some modifications.

Stereo Matching on KITTI - Training

- Pretraining on FlyingThings3D
  - Pretraining for 200 epochs.
  - Batch size – 32.
  - Initial learning rate – 2×10^{-4}.
- Finetuning on KITTI
  - Training is performed for 2000 epochs.
  - Batch size – 16.
  - The initial learning rate is 1×10^{-5}.
    - decayed by 0.5 at the1000th and the1500th epoch.

Stereo matching on KITTI – Results

	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
HD^3S (Ours)	**1.40**	**1.80**	**1.70**	**3.63**	**2.02**	**0.14**

All of the numbers denote percentages of disparity outliers.

Optical Flow

- Network pretrained on FlyingChairs and FlyingThings3D.
- FlyingChairs –
  - batch size – 64.
  - initial learning rate - 4×10^{-4}.
- FlyingThings3D –
  - batch size – 32.
  - initial learning rate - 4×10^{-5}.

Finetuning on Sintel

- Training is performed for 1200 epochs.
- Batch size – 32.
- The initial learning rate is 2×10^{-5}.
    - decayed by 0.5 at the 600th and the 900th epoch.

Optical flow results – Sintel

	Training		Test		Time
Methods	Clean	Final	Clean	Final	(s)
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
MRFlow [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^3F (Ours)	3.84	8.77	-	-	0.08
HD^3F-ft (Ours)	(1.70)	**(1.17)**	4.79	**4.67**	0.08

Average EPE results on MPI Sintel dataset

- "-ft" means finetuning on the Sintel training set
- numbers in the parenthesis are results on data the method has been trained on.

Finetuning on KITTI

- Finetuning on KITTI
  - Training is performed for 2000 epochs.
  - Batch size – 16.
  - The initial learning rate is 1×10^{-5}.
    - decayed by 0.5 at the1000th and the1500th epoch.
- Same parameters as stereo matching.

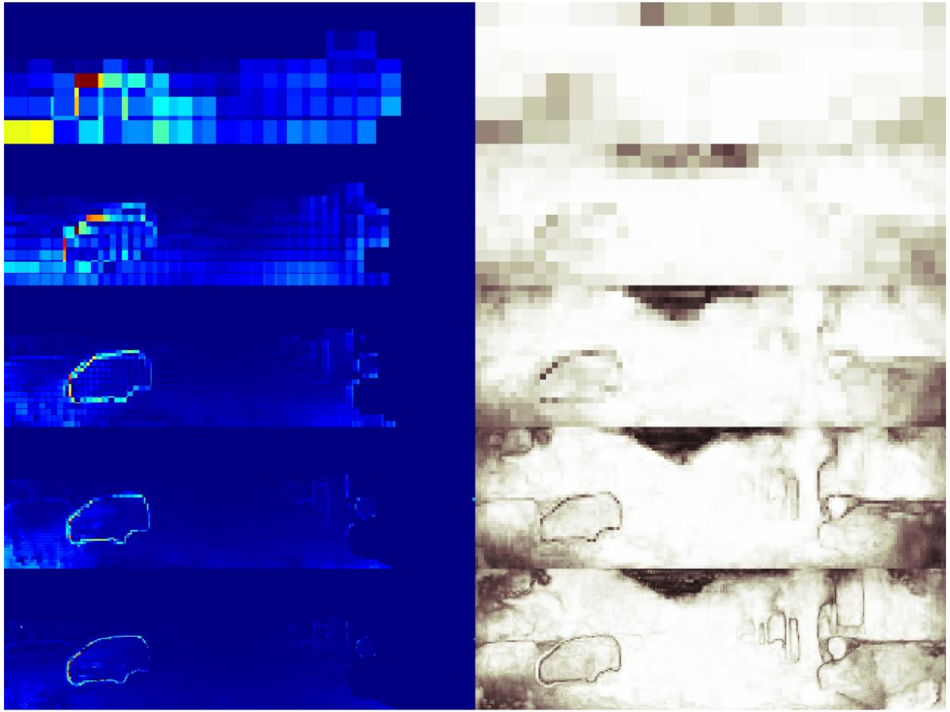
Optical flow results – KITTI

	KITTI 2012			KITTI 2015		
Methods	AEPE *train*	AEPE *test*	F1-Noc *test*	AEPE *train*	F1-all *train*	F1-all *test*
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%**

- "-ft" means fine-tuning on the KITTI training set.
- Numbers in parenthesis are results on data the network has been trained on.

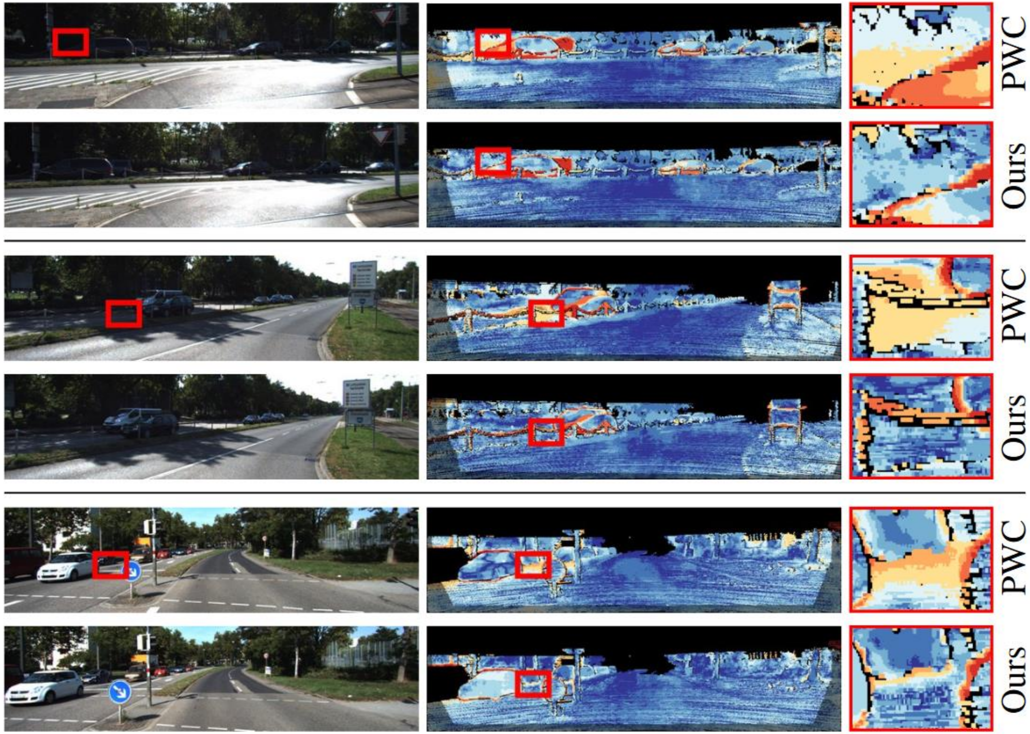
Multiscale predicted flowmaps

Multiscale Error and Confidence maps



Warm color = inaccurate White = more confident

Flow error map comparison with PWC-Net



Summary

- This approach decomposed the match density into multiple scales.
- Learned the decomposed match densities in an end-to-end manner.
- The predicted match densities can be converted into point estimate.
- Provides model-inherent uncertainty measures.

Thank you!