Australian National University

An overview of deep learning on dense matching

Yiran Zhong

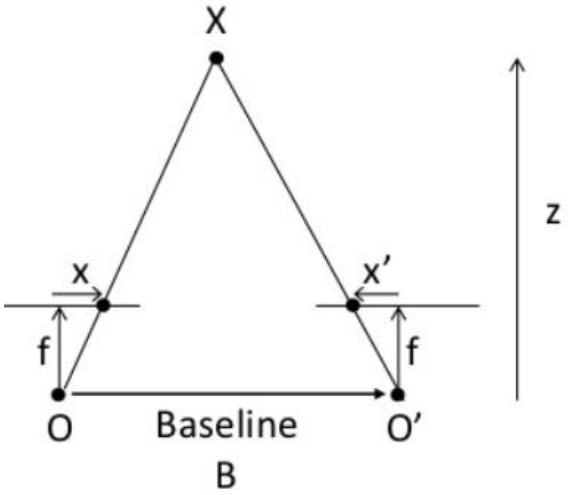
14 August 2020

Contents

- Deep Stereo Matching
  - Problem Definition
  - Methods
  - Future directions
- Deep Optical Flow
  - Problem Definition
  - Methods
  - Future directions

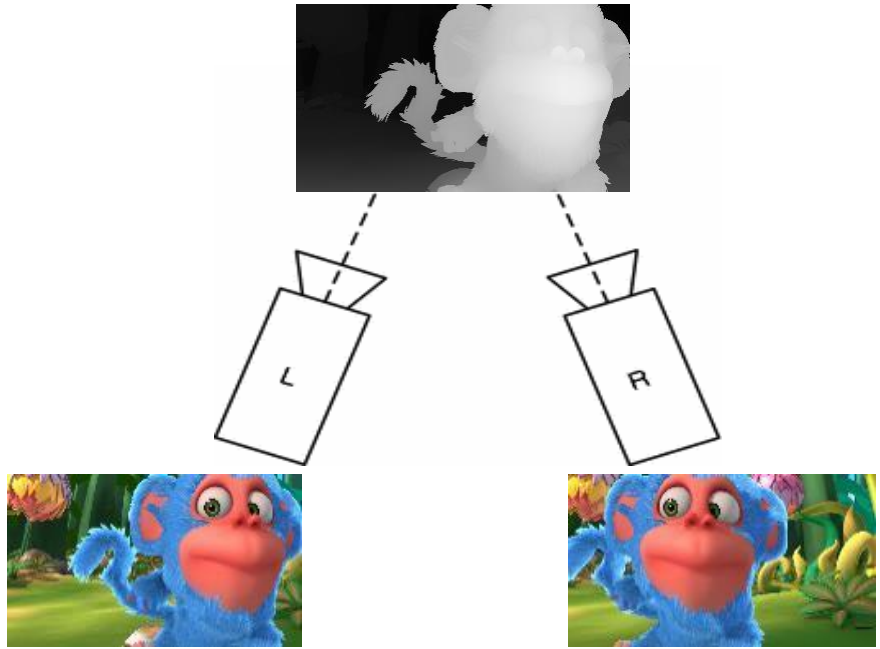
Stereo Matching

Object: *finding dense correspondences for pixels in two rectified images.*



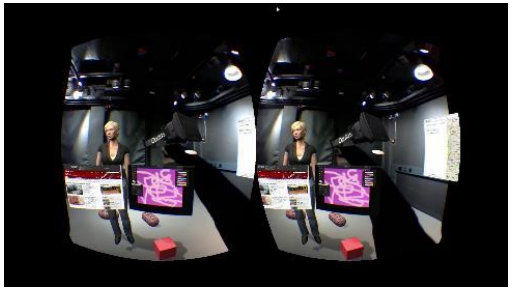
$$disparity = x - x' = \frac{B \cdot f}{z}$$

Disparity is inversely proportional to depth!



Applications

- Robot navigation
- Human robot interface, entertainment
- Virtual / augmented reality
- Autonomous driving



Assumptions

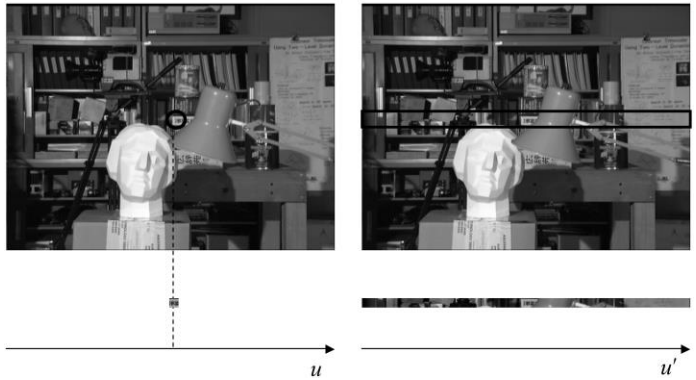
- Color constancy
  - Lambertian surface assumption.
- Epipolar geometry
  - Scanline as epipolar line for rectified pair.
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image.
- Ordering
  - Corresponding points should be in the same order in both views.
- Smoothness
  - Disparities change slowly (in most parts)

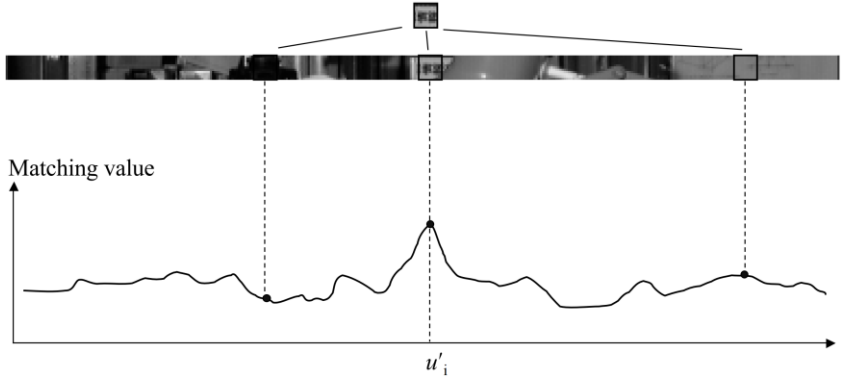
Datasets

- Middlebury datasets
- KITTI Stereo datasets
- ETH 3D datasets
- Freiburg Sceneflow datasets

General Pipeline

- Matching cost computation;
- Cost aggregation;
- Disparity computation / optimization;
- Disparity refinement





Taxonomy

- Learning better features or metrics
- Learning better optimizer
- Learning better refinement
- Direct disparity regression
- Learning better cost aggregation
- Self-supervised deep stereo
- NAS Stereo

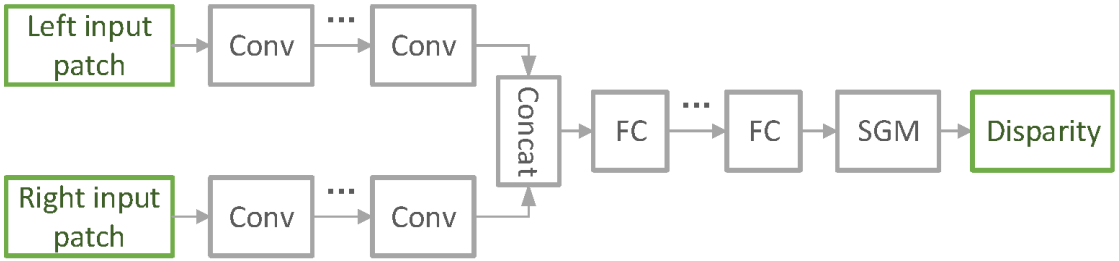
Taxonomy

- Learning better features or metrics
- Learning better optimizer
- Learning better refinement
- Direct disparity regression
- Learning better cost aggregation
- Self-supervised deep stereo
- NAS Stereo

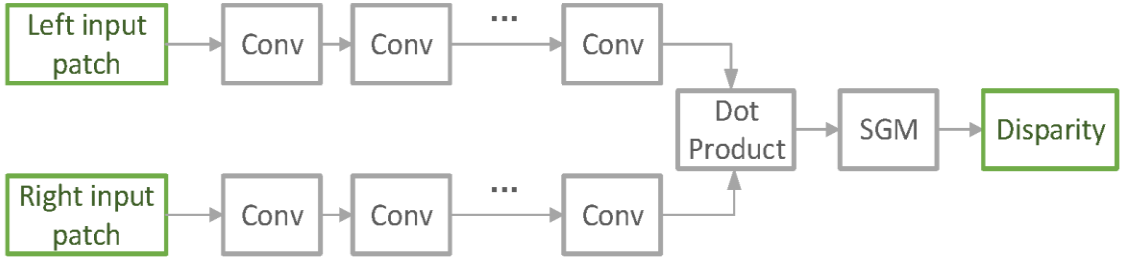
Learning better features or metrics

- MC-CNN (Zbontar & LeCun CVPR 2015, JMLR 2016)
  - Positive / Negative examples for ‘matching’ and ‘non-matching’.
  - Two architectures proposed:
    - MC-CNN-acrt
    - MC-CNN-fast
  - A set of post-processing steps:
    - Semiglobal Matching
    - Occlusion interpolation
    - Subpixel enhancement
    - Color-weighted median filter
  - Trained features can be applied to conventional global algorithms: LocalExp, MeshStereo …

Network Architecture

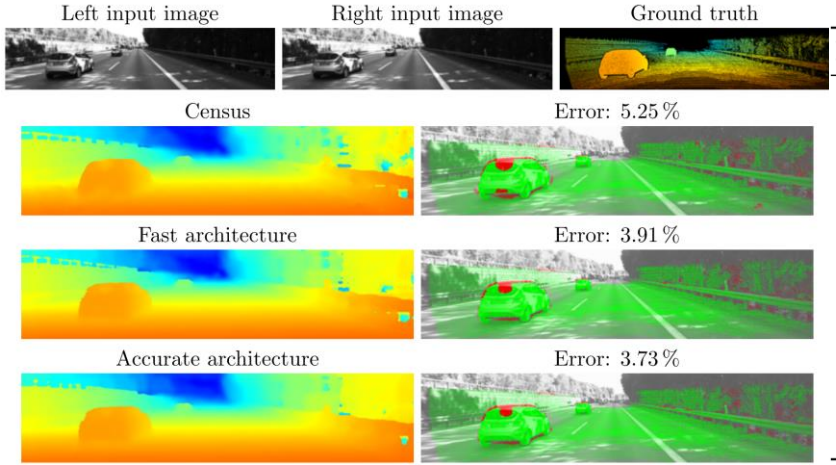


MC-CNN-acrt



MC-CNN-fast

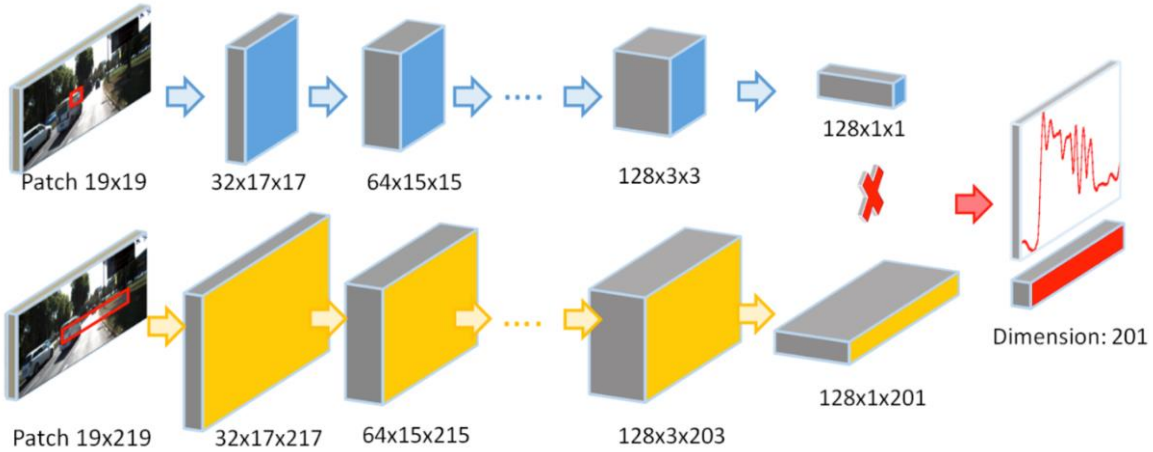
Performance on KITTI 2015



Rank	Method		Setting	Error	Runtime
1	**MC-CNN-acrt**	**Accurate architecture**		3.89	67
	MC-CNN-fst	**Fast architecture**		4.62	0.8
2	SPS-St	Yamaguchi et al. (2014)		5.31	2
3	OSF	Menze and Geiger (2015)	F	5.79	3000
4	PR-Sceneflow	Vogel et al. (2013)	F	6.24	150
5	SGM+C+NL	Hirschmüller (2008); Sun et al. (2014)	F	6.84	270
6	SGM+LDOF	Hirschmüller (2008); Brox and Malik (2011)	F	6.84	86
7	SGM+SF	Hirschmüller (2008); Hornacek et al. (2014)	F	6.84	2700
8	ELAS	Geiger et al. (2011)		9.72	0.3
9	OCV-SGBM	Hirschmüller (2008)		10.86	1.1
10	SDM	Kostková and Sára (2003)		11.96	60

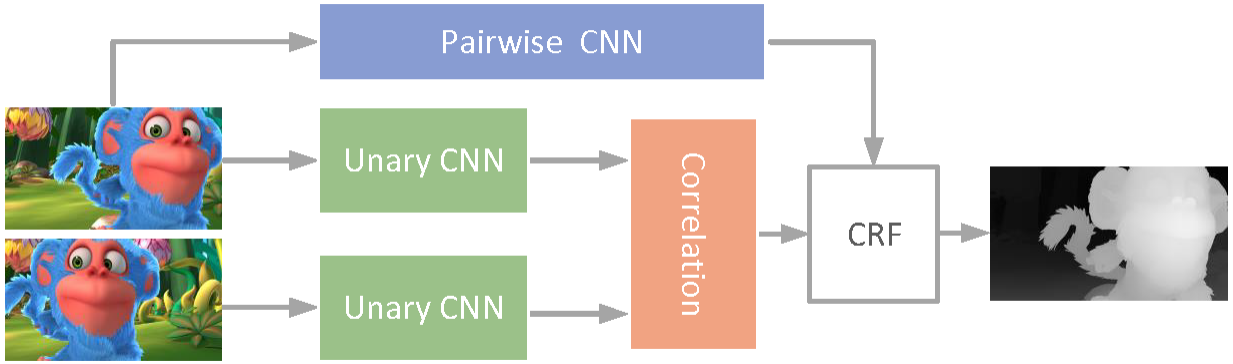
Learning better features or metrics

- A deep embedding model for stereo matching cost. (Chen et.al. ICCV 2015)
- Efficient deep stereo matching (Luo et.al. CVPR 2016)
  - Smaller Siamese network
  - Use multi-class classification loss instead of binary classification loss



Learning better features or metrics

- Hybrid CNN-CRF (Knobelreiter et.al. CVPR 2017)
  - Unary-CNN for unary cost
  - Pairwise-CNN for pairwise cost
  - End-to-end training in discriminative formulation (structured SVM, bilevel optimization)
  - No post-processing needed



Learning better features or metrics

- Weakly supervised learning of deep metrics for stereo. (Tulyakov etal. ICCV 2017)
  - Allow MC-CNN to be trained under a weakly supervised manner.
  - Triplet
    - *reference patches* $\mathbf{e}^r = (p_1^r, p_2^r, .., p_W^r)$ extracted from a horizontal line of a left rectified stereo image,
    - *positive patches* $\mathbf{e}^+ = (p_1^+, p_2^+, .., p_W^+)$ extracted from the corresponding horizontal in the right rectified stereo image, and
    - *negative patches* $\mathbf{e}^- = (p_1^-, p_2^-, .., p_W^-)$ extracted from another horizontal line of a right rectified stereo image,
  - For every patch, the similarity of the best match should be greater than the similarity of the second-best match.

Taxonomy

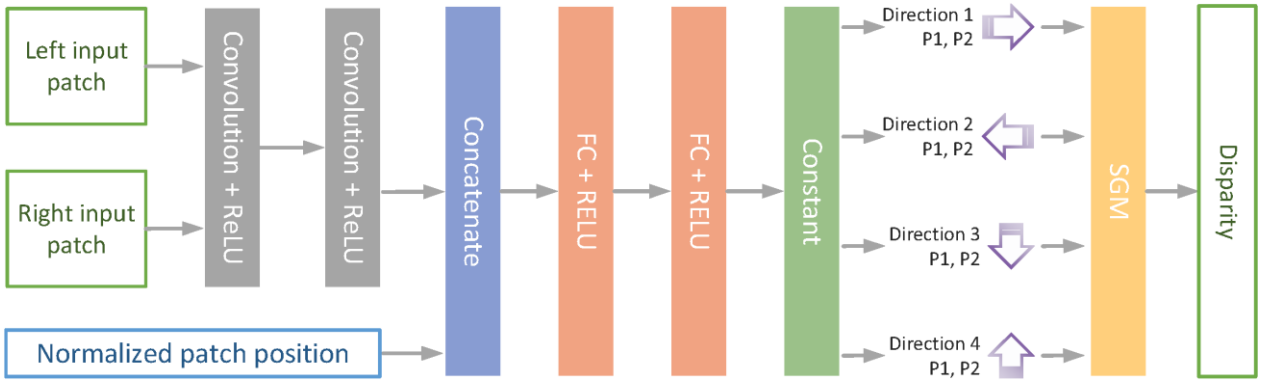
- Learning better features or metrics
- Learning better optimizer
- Learning better refinement
- Direct disparity regression
- Learning better cost aggregation
- Self-supervised deep stereo
- NAS Stereo

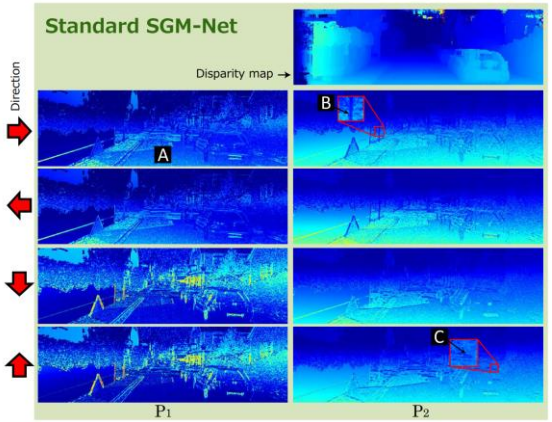
Learning better optimizer

- SGM-Net. (Seki etal. CVPR 2017)
  - A ‘deep’ implementation of SGM.
  - Recall SGM energy function:

$$\mathbf{E} = \sum_{\mathbf{x}} (C(\mathbf{x}, d^{\mathbf{x}}) + \sum_{\mathbf{y} \in \mathbf{N_x}} p_1 T[|d^{\mathbf{x}} - d^{\mathbf{y}}| = 1] + \sum_{\mathbf{y} \in \mathbf{N_x}} p_2 T[|d^{\mathbf{x}} - d^{\mathbf{y}}| > 1])$$

where $C(\mathbf{x}, d^{\mathbf{x}})$ is a matching cost at pixel $\mathbf{x}$ of disparity $d^{\mathbf{x}}$. P_1 represents a penalty for all pixels $\mathbf{y}$ in the neighbourhood $\mathbf{N_x}$ of $\mathbf{x}$, while p_2 is a penalty for discontinuousness. $T(\cdot)$ is Kronecker delta function which gives 1 when a condition in the bracket is satisfied.





Performance on KITTI 2015

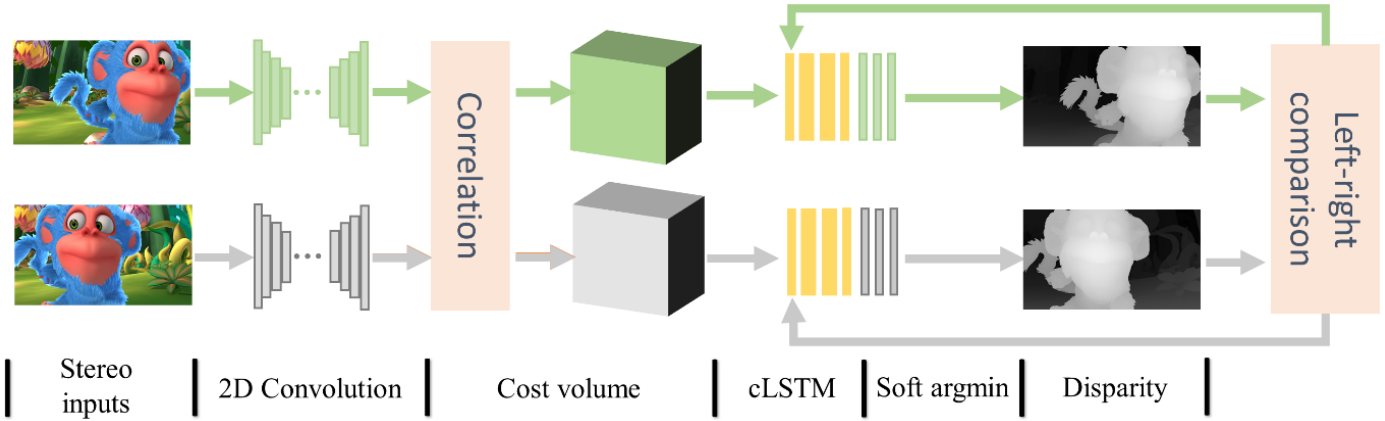
Rank	Method	D1-bg	D1-fg	D1-all
1	Standard SGM-Net	**2.23%**	7.44%	**3.09%**
1	Displets v2 [12]	2.73%	4.95%	**3.09%**
3	PBCP [28]	2.27%	7.72%	3.17%
4	Signed SGM-Net	2.24%	7.94%	3.18%
5	MC-CNN-acrt [38]	2.48%	7.64%	3.33%
7	DispNetC [22]	4.11%	**3.72%**	4.05%

Taxonomy

- Learning better features or metrics
- Learning better optimizer
- Learning better refinement
- Direct disparity regression
- Learning better cost aggregation
- Self-supervised deep stereo
- NAS Stereo

Learning better refinement

- Patch based confidence prediction for dense disparity map.(Seki et al. BMVC 2016)
- Detect, replace, refine, deep structured prediction for pixel wise labeling.(Spyros et al. CVPR 2017)
- LRCR-Net. (Zequn et al. CVPR 2018)

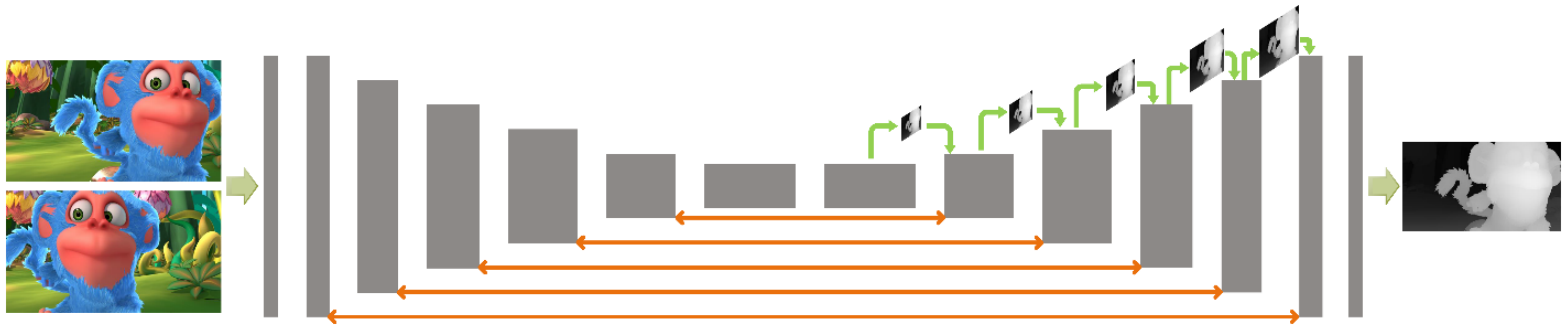


Taxonomy

- Learning better features or metrics
- Learning better optimizer
- Learning better refinement
- Direct disparity regression
- Learning better cost aggregation
- Self-supervised deep stereo
- NAS Stereo

Direct disparity regression

- DispNet. (Mayer et al. CVPR 2016)
  - The first end-to-end stereo matching network.
  - Purely data driven
  - Easy to attack
  - Following work:
    - iResNet. (Liang et al. CVPR 2018) Winner of Robust Vision Challenge
    - Occlusions, Motion and Depth Boundaries with a Generic Network for Optical Flow, Disparity, or Scene Flow Estimation. (Ilg et al. ECCV 2018)

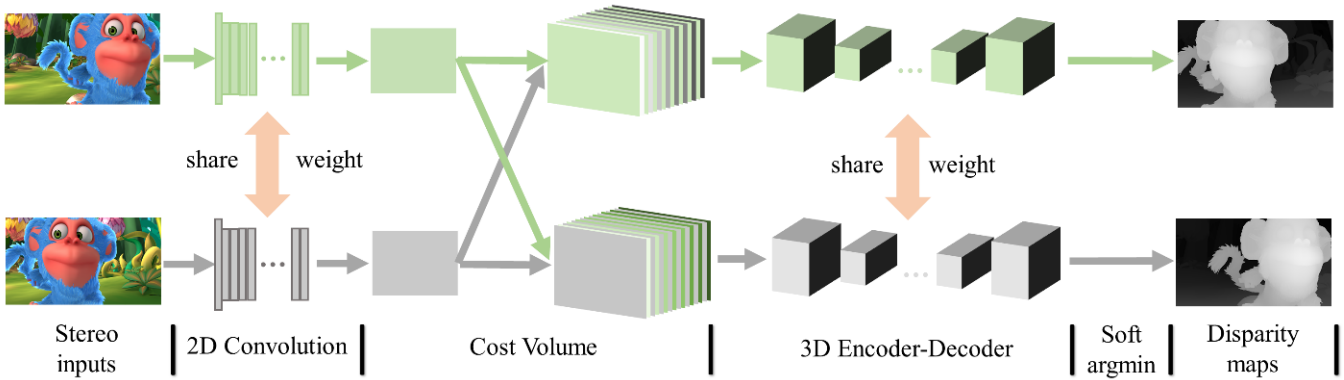


Taxonomy

- Learning better features or metrics
- Learning better optimizer
- Learning better refinement
- Direct disparity regression
- Learning better cost aggregation
- Self-supervised deep stereo
- NAS Stereo

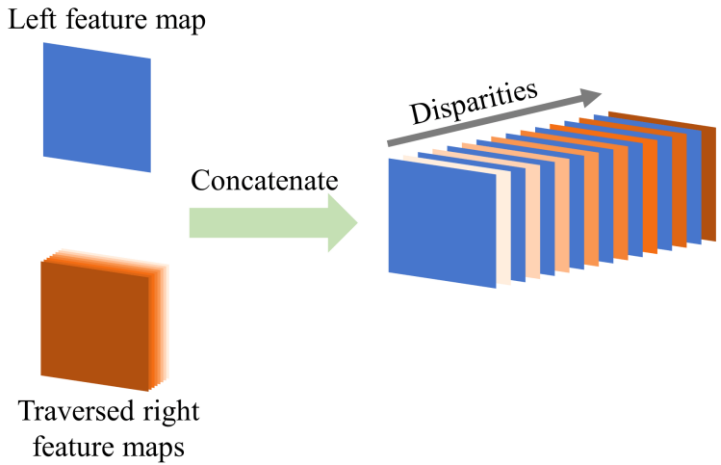
Learning better cost aggregation

- GC-Net. (Kendall et al. CVPR 2017)
  - Introduced 3D convolutions to learning cost aggregation
  - Robust to attack
  - Following work:
    - PSMnet. (Chang et al. CVPR 2018)
    - GANet. (Zhang et al. CVPR 2019)



Learning better cost aggregation

- Feature Volume Construction



- Soft argmin

$$soft\ argmin := \sum_{d=0}^{D_{max}} d \times \sigma(-c_d)$$

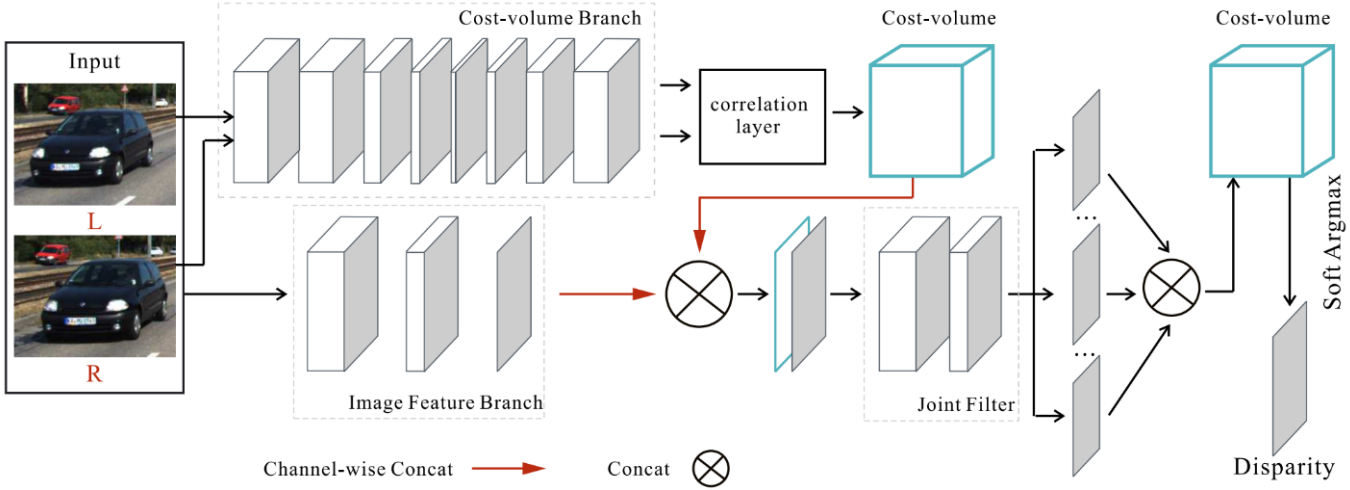
Taxonomy

- Learning better features or metrics
- Learning better optimizer
- Learning better refinement
- Direct disparity regression
- Learning better cost aggregation
- Self-supervised deep stereo
- NAS Stereo

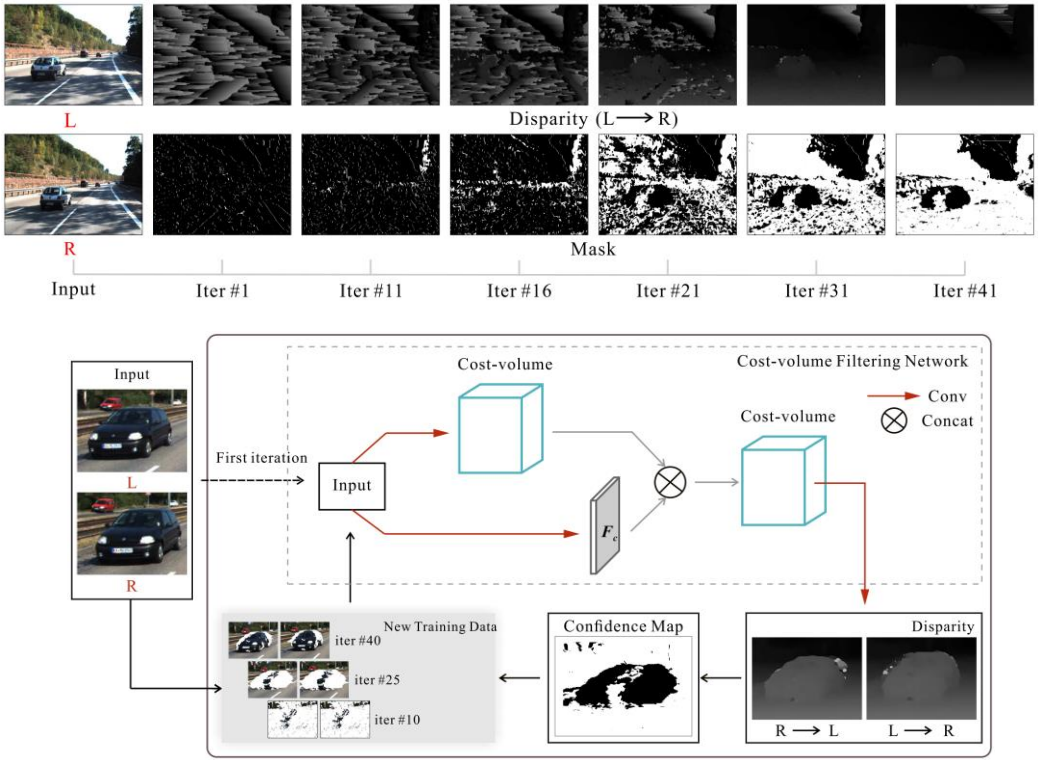
Unsupervised deep stereo matching

- Unsupervised learning of stereo matching. (Zhou et al. ICCV 2017)
  - Idea similar to PatchMatch Stereo:

    Start with a random disparity map, select confident pixels with low warping errors, use the confident pixels' disparities as pseudo ground truth to supervised the network.

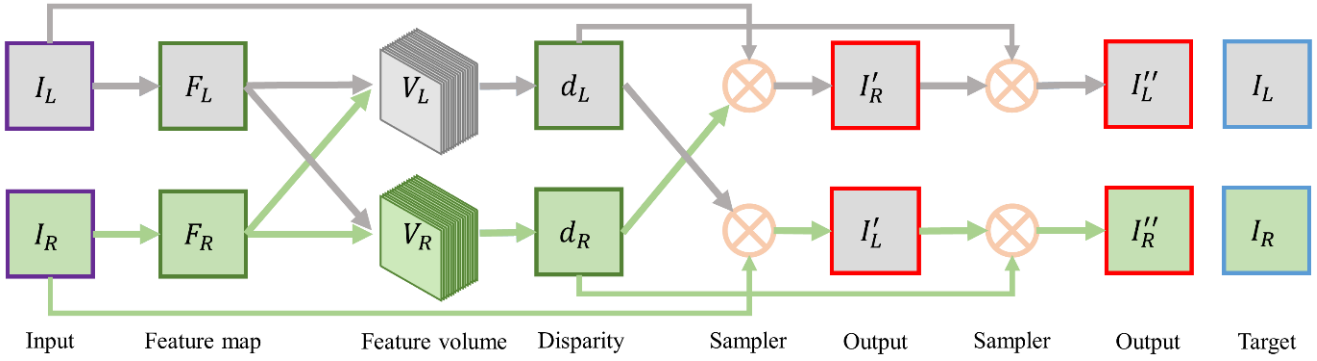


Unsupervised deep stereo matching



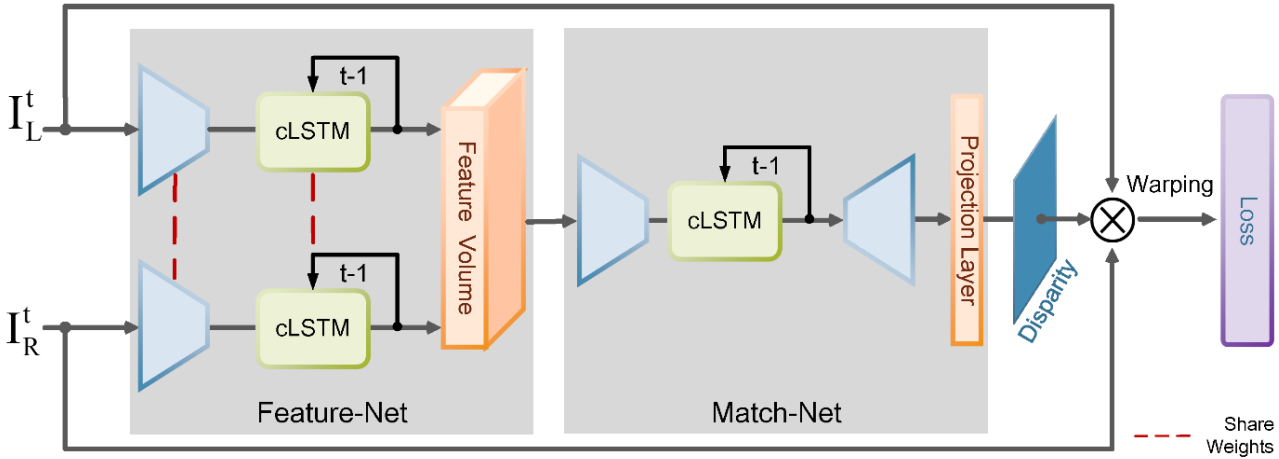
Self-supervised deep stereo matching

- SsSMNet. (Zhong et al. 2017)
  - Minimize photometric warping error.
  - No high-level semantic information used.
  - Following work:
    - Open-world Stereo Video Matching. (Zhong et al. ECCV 2018)
    - Active Stereo Net. (Zhang et al. ECCV 2018)



Self-supervised deep stereo matching

- Open-world Stereo Video Matching. (Zhong et al. ECCV 2018)
  - Naturally exploits *temporal coherency* in stereo video input, via the use of RNN/LSTM.
  - No need ground-truth disparity as supervision, which makes it generalize well to unseen and new scenarios.
  - Able to learn *on-the-fly*.



Open-world Stereo Video Matching

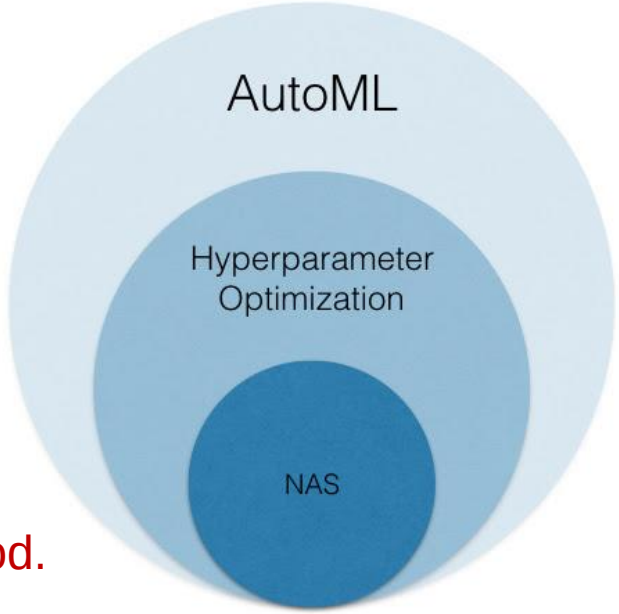
- Self-adapting training and testing
  - Dynamically evolve in time.
  - The network has memory units (i.e. the two LSTM blocks), which enable the network to adjust its current behavior (partly) based its past experiences;
  - Always run an online back-propagation updating procedure after any feed-forward process.

Taxonomy

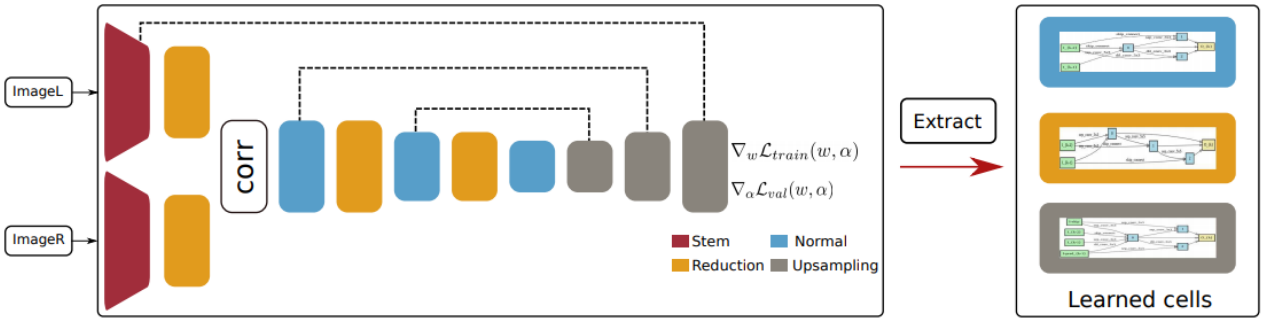
- Learning better features or metrics
- Learning better optimizer
- Learning better refinement
- Direct disparity regression
- Learning better cost aggregation
- Self-supervised deep stereo
- NAS Stereo

NAS Stereo

- AutoDispNet: Improving Disparity Estimation With AutoML (ICCV19)
  - DispNet-like architecture
  - Using DARTS to search Cell Structure.
  - Leveraging Bayesian optimization to do hyperparameter search



DARTS is a Differentiable Architecture Search Method.



(a) Search network

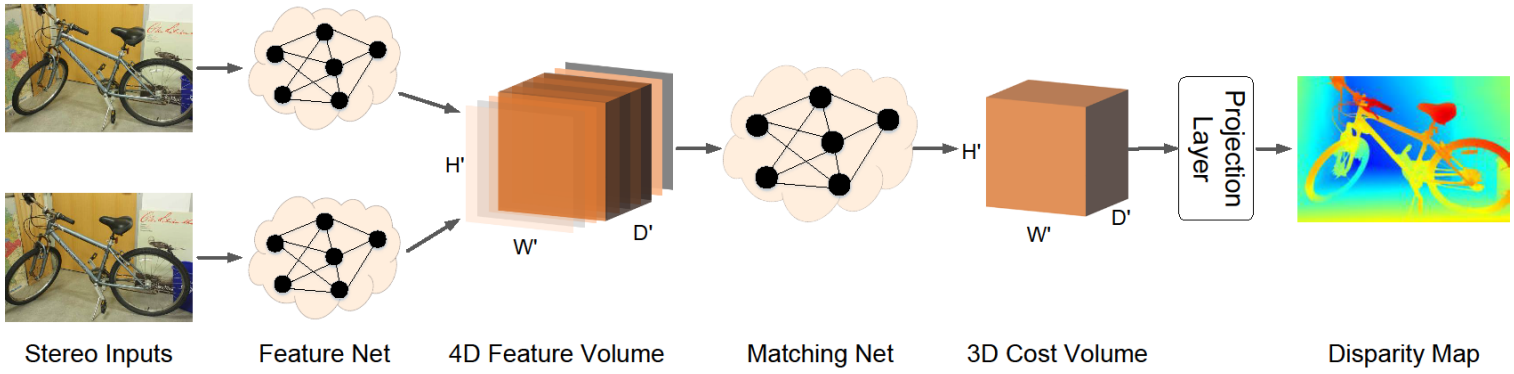
(b) AutoDispNet-CSS

NAS Stereo

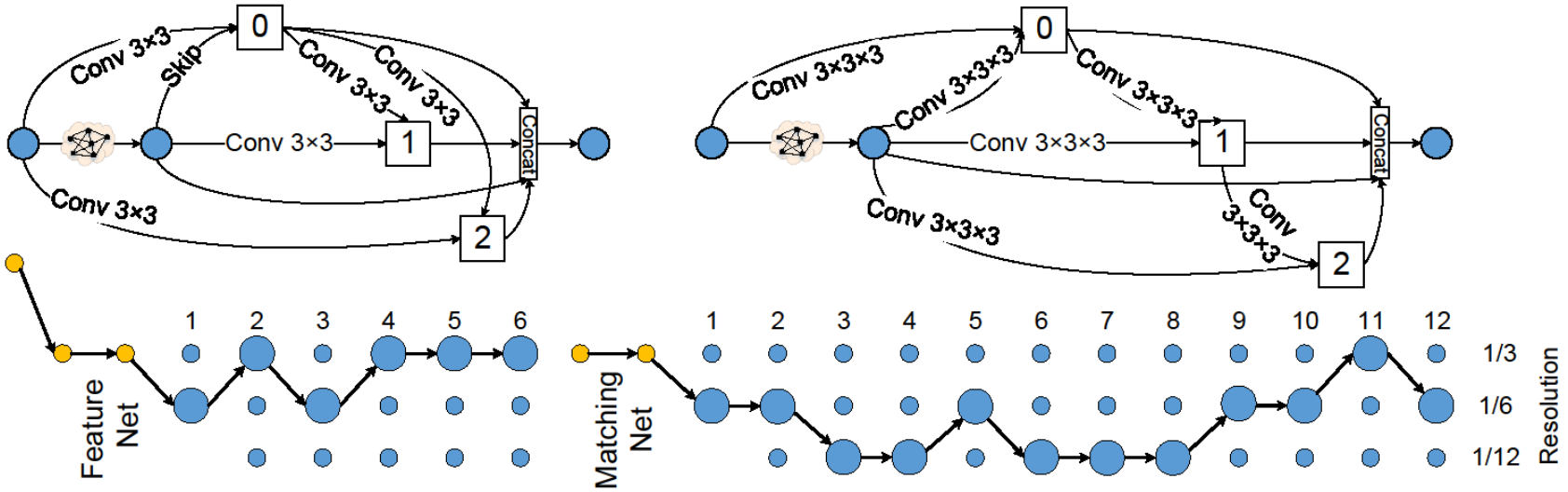
- LEAStereo
  - Full Architecture Search
  - Dominating KITTI 12, KITTI 15, Middlebury benchmarks.

	Search Level	Params	KITTI 2012	KITTI 2015	Runtime
AutoDispNet	Cell	111M	1.70%	2.18%	0.9 s
LEAStereo	Full Network	1.8M	1.13%	1.65%	0.3 s

Table 1: AutoDispNet *vs* LEAStereo



Searched architecture



KITTI 2012 Stereo benchmark

Table: All ▾ Error threshold: 3 pixels ▾ Evaluation area: All pixels ▾

	Method	Setting	Code	Out-Noc	Out-All	Avg-Noc	Avg-All	Density	Runtime	Environment	Compare
1	FDCVNet			0.00 %	0.00 %	0.0 px	0.0 px	0.00 %	0.03 s	1 core @ 2.5 Ghz (C/C++)	
2	attention_global_net			0.00 %	0.00 %	0.0 px	0.0 px	0.00 %	1.5 s	4 cores @ 2.5 Ghz (Python)	
ERROR: Wrong syntax in BIBTEX file.											
3				0.00 %	0.00 %	0.0 px	0.0 px	0.00 %			
4	LEAStereo			1.13 %	1.45 %	0.5 px	0.5 px	100.00 %	0.3 s	GPU @ 2.5 Ghz (Python)	
5	MSMD-Net			1.14 %	1.47 %	0.4 px	0.5 px	100.00 %	0.72 s	1 core @ 2.5 Ghz (C/C++)	

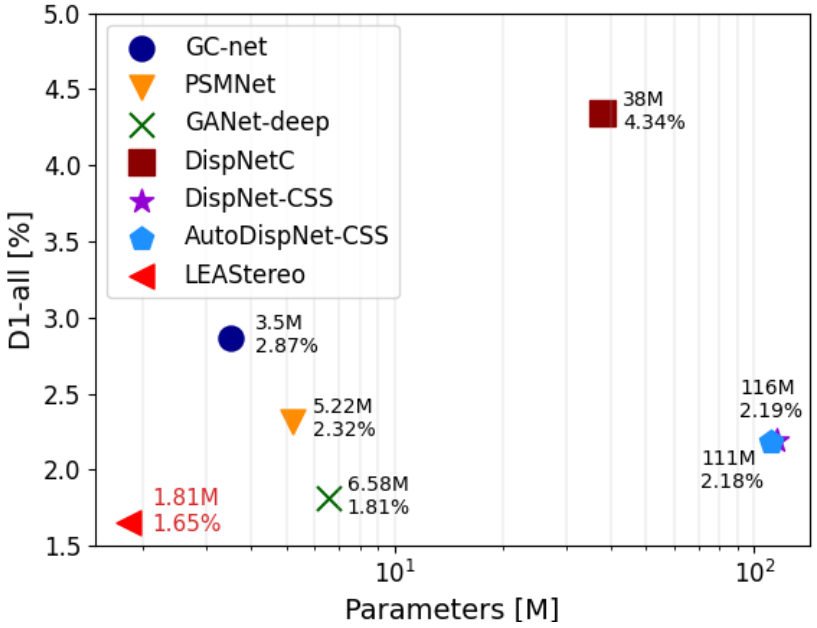
KITTI 2015 Stereo benchmark

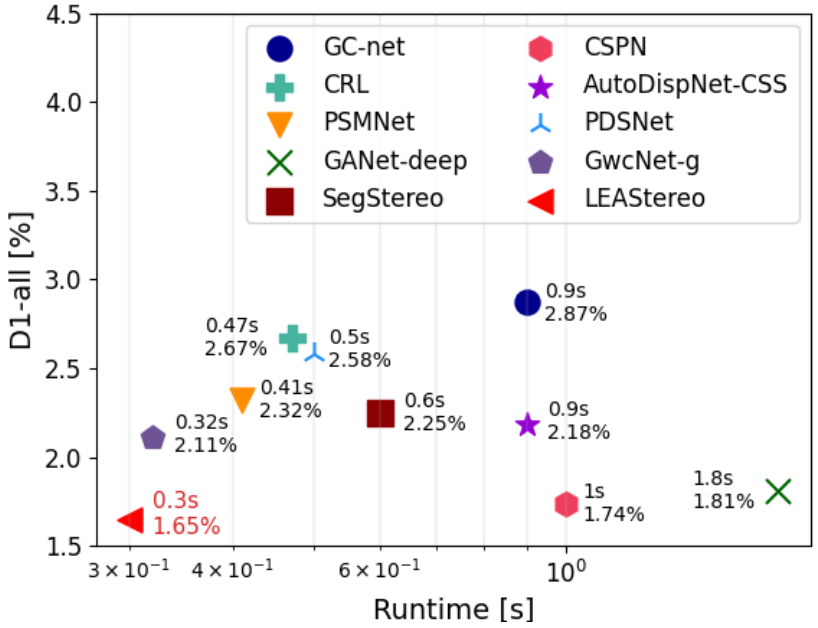
Evaluation ground truth: All pixels ▾ Evaluation area: All pixels ▾

	Method	Setting	Code	D1-bg	D1-fg	D1-all	Density	Runtime	Environment	Compare
1	LEAStereo			1.40 %	2.91 %	1.65 %	100.00 %	0.30 s	GPU @ 2.5 Ghz (Python)	
2	Dahua_Stereo			1.48 %	2.83 %	1.71 %	100.00 %	1.52 s	GPU @ 2.5 Ghz (Python)	
3	MSMD-Net(only MS)			1.41 %	3.22 %	1.71 %	100.00 %	0.52 s	1 core @ 2.5 Ghz (C/C++)	
4	CSPN			1.51 %	2.88 %	1.74 %	100.00 %	1.0 s	GPU @ 2.5 Ghz (Python)	

X. Cheng, P. Wang and R. Yang: Learning Depth with Convolutional Spatial Propagation Network. IEEE Transactions on Pattern Analysis and Machine Intelligence(T-PAMI) 2019.

KITTI 2015


GC-net
PSMNet
GANet-deep
DispNetC
DispNet-CSS
AutoDispNet-CSS
LEAStereo
38M
4.34%
3.5M
2.87%
5.22M
2.32%
116M
2.19%
111M
2.18%
1.81M
1.65%
6.58M
1.81%
D1-all [%]
Parameters [M]


GC-net
CRL
PSMNet
GANet-deep
SegStereo
CSPN
AutoDispNet-CSS
PDSNet
GwcNet-g
LEAStereo
0.9s
2.87%
0.47s
2.67%
0.5s
2.58%
0.41s
2.32%
0.6s
2.25%
0.32s
2.11%
0.9s
2.18%
1s
1.74%
1.8s
1.81%
0.3s
1.65%
D1-all [%]
Runtime [s]

MIDDLEBURY 2014

Set: test dense test sparse training dense training sparse
Metric: bad 0.5 bad 1.0 bad 2.0 bad 4.0 avgerr rms A50 A90 A95 A99 time time/MP time/GD
Mask: nonocc all
plot selected show invalid Reset sort Reference list

bad 4.0 (%)

Date	Name	Res	Weight Avg	Austr MP: 5.6 nd: 290 im0 im1 GT nonocc	AustrP MP: 5.6 nd: 290 im0 im1 GT nonocc	Bicyc2 MP: 5.6 nd: 250 im0 im1 GT nonocc	Class MP: 5.7 nd: 610 im0 im1 GT nonocc	ClassE MP: 5.7 nd: 610 im0 im1 GT nonocc	Compu MP: 1.5 nd: 256 im0 im1 GT nonocc	Crusa MP: 5.5 nd: 800 im0 im1 GT nonocc	CrusaP MP: 5.5 nd: 800 im0 im1 GT nonocc	Djemb MP: 5.7 nd: 320 im0 im1 GT nonocc	DjembL MP: 5.7 nd: 320 im0 im1 GT nonocc	Hoops MP: 5.7 nd: 410 im0 im1 GT nonocc	Livgrm MP: 5.9 nd: 320 im0 im1 GT nonocc	Nkuba MP: 5.5 nd: 570 im0 im1 GT nonocc	Plants MP: 5.6 nd: 320 im0 im1 GT nonocc	Stairs MP: 5.2 nd: 450 im0 im1 GT nonocc
05/28/20	LEAStereo	H	**2.75** 1	3.57 14	2.46 6	1.68 3	1.60 6	3.47 4	1.87 5	**1.57** 1	1.59 5	1.44 3	4.02 3	**4.11** 1	5.01 3	**5.13** 1	**2.89** 1	**3.22** 1
05/26/18	NOSS_ROB	H	3.46 2	2.69 4	2.37 4	1.99 7	1.19 2	**3.11** 1	1.75 3	1.63 5	**1.26** 1	1.87 15	5.60 5	4.63 3	5.66 4	10.4 19	3.60 9	7.02 22
03/09/19	3DMST-CM	H	3.57 3	3.07 10	2.77 11	1.80 4	1.79 10	5.43 16	2.58 10	1.71 7	2.01 15	1.48 4	6.86 9	5.14 5	**4.41** 1	8.85 8	3.84 12	6.29 12

Set: test dense test sparse training dense training sparse
Metric: bad 0.5 bad 1.0 bad 2.0 bad 4.0 avgerr rms A50 A90 A95 A99 time time/MP time/GD
Mask: nonocc all
plot selected show invalid Reset sort Reference list

bad 4.0 (%)

Date	Name	Res	Weight Avg	Austr MP: 5.6 nd: 290 im0 im1 GT nonocc	AustrP MP: 5.6 nd: 290 im0 im1 GT nonocc	Bicyc2 MP: 5.6 nd: 250 im0 im1 GT nonocc	Class MP: 5.7 nd: 610 im0 im1 GT nonocc	ClassE MP: 5.7 nd: 610 im0 im1 GT nonocc	Compu MP: 1.5 nd: 256 im0 im1 GT nonocc	Crusa MP: 5.5 nd: 800 im0 im1 GT nonocc	CrusaP MP: 5.5 nd: 800 im0 im1 GT nonocc	Djemb MP: 5.7 nd: 320 im0 im1 GT nonocc	DjembL MP: 5.7 nd: 320 im0 im1 GT nonocc	Hoops MP: 5.7 nd: 410 im0 im1 GT nonocc	Livgrm MP: 5.9 nd: 320 im0 im1 GT nonocc	Nkuba MP: 5.5 nd: 570 im0 im1 GT nonocc	Plants MP: 5.6 nd: 320 im0 im1 GT nonocc	Stairs MP: 5.2 nd: 450 im0 im1 GT nonocc
05/28/20	LEAStereo	H	**6.33** 1	4.59 7	3.55 2	**3.16** 1	**5.63** 1	**7.77** 1	**8.46** 1	5.86 4	**4.77** 1	**2.97** 1	5.94 3	**13.6** 1	9.11 4	**9.13** 1	**7.14** 1	**6.83** 1
03/09/19	3DMST-CM	H	8.10 2	4.66 8	4.36 10	4.35 2	6.89 5	11.2 9	11.5 6	5.51 2	6.09 4	3.35 3	8.77 9	14.8 4	**6.93** 1	15.2 9	9.87 5	15.1 16
05/26/18	NOSS_ROB	H	8.37 3	3.96 2	3.62 4	5.01 6	6.62 2	8.72 2	10.6 3	7.53 10	8.34 13	3.60 9	7.71 5	14.5 3	8.42 3	16.5 14	10.2 6	13.5 9

Future directions

- Efficient stereo matching.
  - For autonomous driving, it requires less than 3ms per pair.
  - AutoML
- Domain adaptation
- Long range, small baseline.

Australian National University

An overview of deep learning on dense matching

Yiran Zhong

21 August 2020

Contents

- Deep Stereo Matching
  - Problem Definition
  - Methods
  - Future directions
- Deep Optical Flow
  - Problem Definition
  - Methods
  - Future directions

Optical flow

The apparent movements of pixels between two consecutive frames.

Frame 1 | Frame 2 | Flow Vectors | Flow filed color visualization

Assumptions

- Color constancy
  - Lambertian surface assumption.
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image.
- Smoothness
  - Flow changes slowly (in most parts)
- Small motion

Basic Constraint

Color constancy assumption

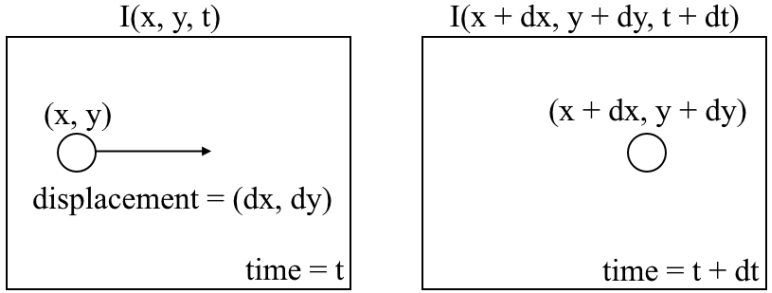
$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

Taylor Series Approximation

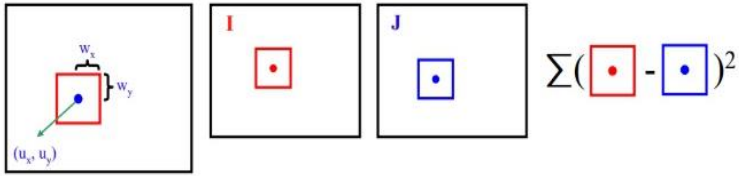
$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + \ldots$$

$$\Rightarrow \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0$$



Smoothness assumption



Datasets

- Flying Chairs Dataset
- Scene Flow Dataset
- MPI Sintel Dataset
- VIPER Dataset
- Middlebury Dataset
- KITTI Dataset

Synthetic Datasets

Real Datasets

Taxonomy

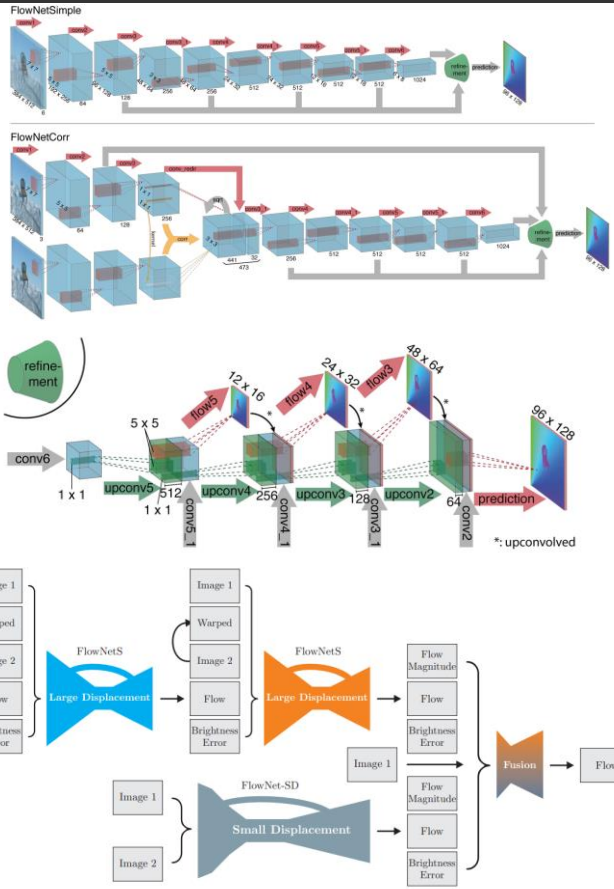
- Direct regression
- Learning better features
- Learning better cost aggregation
- Learning better cost metrics
- Learning better refinement
- Self-supervised deep optical flow

Taxonomy

- Direct regression
- Learning better features
- Learning better cost aggregation
- Learning better cost metrics
- Learning better refinement
- Self-supervised deep optical flow

Direct regression

- FlowNet (ICCV 2015)
  - The 1^{th} end-to-end method
  - Brutal-force
- FlowNet2 (CVPR 2017)
  - Training strategy is important
  - Multi-stage refinement
- FlowNet3 (ECCV18)
  - Occlusions and residual connections



Taxonomy

- Direct regression
- Learning better features
- Learning better cost aggregation
- Learning better cost metrics
- Learning better refinement
- Self-supervised deep optical flow

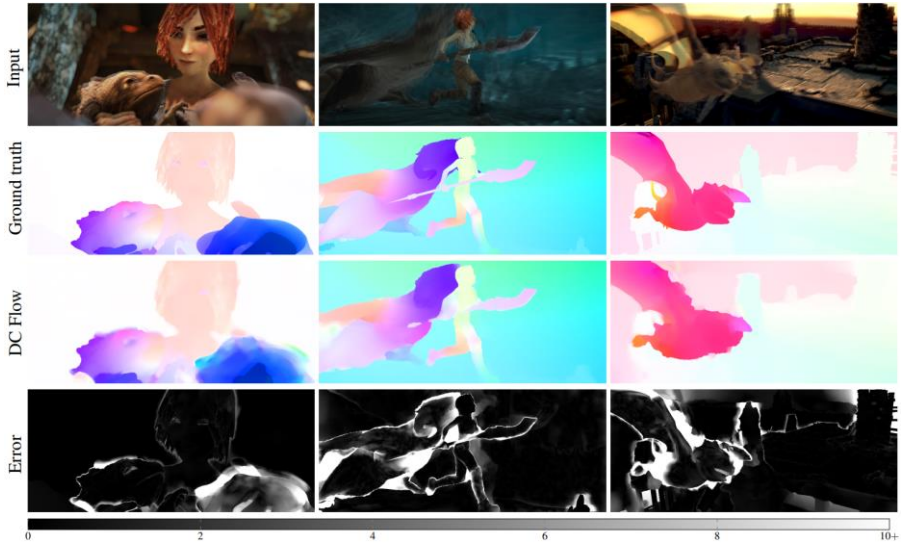
Learning better features

- DCFlow (CVPR 2017)
  - Introducing stereo pipeline to optical flow
  - Learned feature embedding – like MC-CNN
  - Build a 4D cost volume with dot product (HxWxUxV)
  - Regularize cost volume with modified SGM
  - Post-processing

Unable to deal dramatic occlusion, strong motion blur or large motion of untextured objects.

Learning better features

- DCFlow (CVPR 2017)



Method	all	noc	occ	d0-10	d10-60	d60-140	s0-10	s10-40	s40+
PatchBatch [12]	6.783	3.507	33.498	6.080	3.408	2.103	**0.725**	**3.064**	45.858
EpicFlow [27]	6.285	3.060	32.564	5.205	2.611	2.216	1.135	3.727	38.021
DiscreteFlow [25]	6.077	2.937	31.685	5.106	2.459	1.945	1.074	3.832	36.339
CPM-Flow [18]	5.960	2.990	30.177	5.038	2.419	2.143	1.155	3.755	35.136
FullFlow [7]	5.895	2.838	30.793	4.905	2.506	1.913	1.136	3.373	35.592
SPM-BPv2 [21]	5.812	2.754	30.743	4.736	2.255	1.933	1.048	3.468	35.118
DDF [13]	5.728	2.623	31.042	5.347	2.478	1.590	0.959	3.072	35.819
FlowFields+ [3]	5.707	2.684	30.356	4.691	2.117	1.793	1.131	3.330	34.167
DC Flow	**5.119**	**2.283**	**28.228**	**4.665**	**2.108**	**1.440**	1.052	3.434	**29.351**

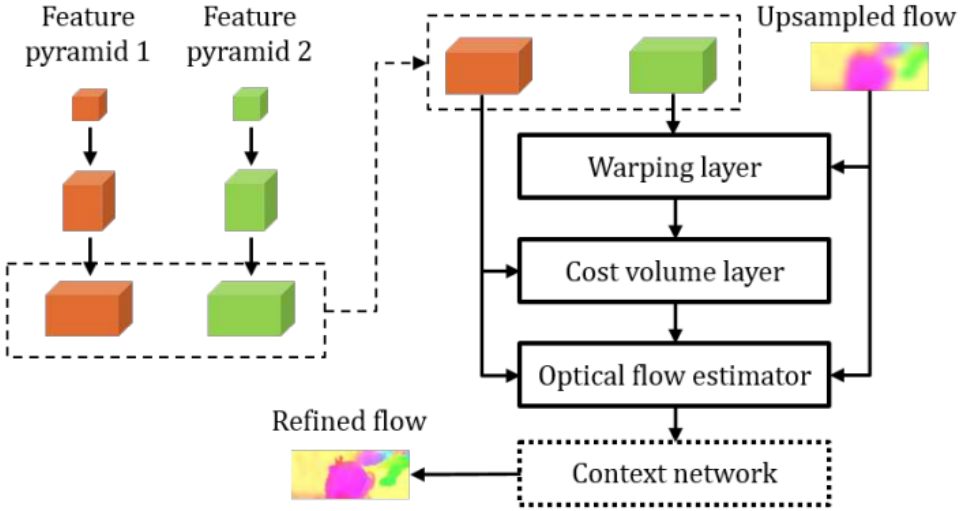
Method	Domain-agnostic	Non-occluded pixels (%)			All pixels (%)			Runtime
		Fl-bg	Fl-fg	Fl-all	Fl-bg	Fl-fg	Fl-all	
SOF [31]	✗	8.11	18.16	9.93	14.63	22.83	15.99	6 min
JFS [19]	✗	7.85	14.97	9.14	15.90	19.31	16.47	13 min
SDF [2]	✗	5.75	18.38	8.04	8.61	23.01	11.01	–
EpicFlow [27]	✓	15.00	24.34	16.69	25.81	28.69	26.29	15 sec
FullFlow [7]	✓	12.97	20.58	14.35	23.09	24.79	23.57	4 min
CPM-Flow [18]	✓	12.77	18.71	13.85	22.32	22.81	22.40	4.2 sec
DiscreteFlow [25]	✓	9.96	**17.03**	11.25	21.53	**21.76**	21.57	3 min
DDF [13]	✓	10.44	21.32	12.41	20.36	25.19	21.17	1 min
PatchBatch [12]	✓	10.06	22.29	12.28	19.98	26.50	21.07	50 sec
DC Flow	✓	**8.04**	19.84	**10.18**	**13.10**	23.70	**14.86**	8.6 sec

Taxonomy

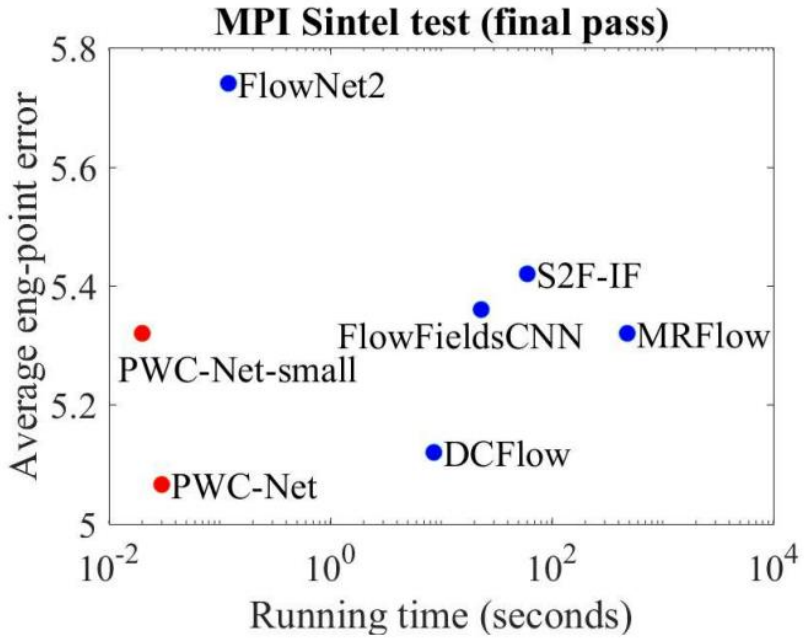
- Direct regression
- Learning better features
- Learning better cost aggregation
- Learning better cost metrics
- Learning better refinement
- Self-supervised deep optical flow

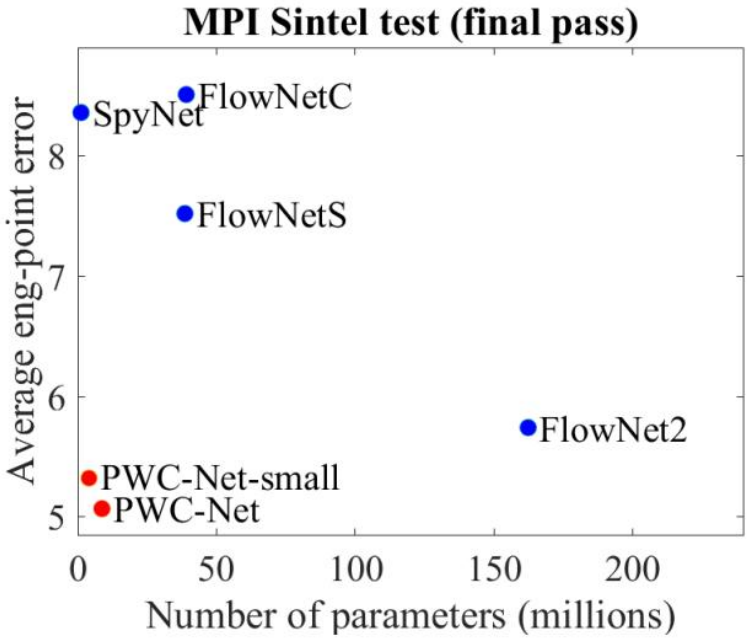
Learning better cost aggregation

- PWCNet (CVPR 2018)
  - Pyramidal processing
  - Warping
  - Cost volume
  - Sophisticated training scheme



Learning better cost aggregation



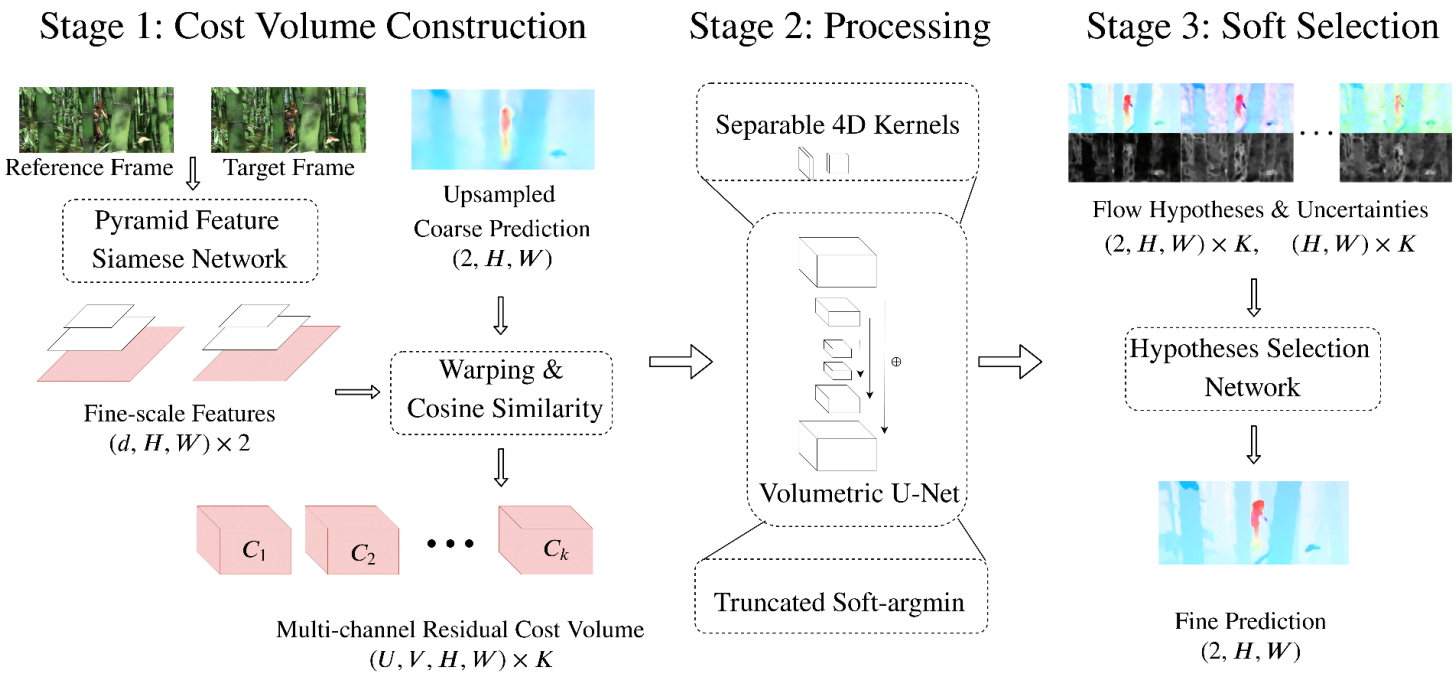


Learning better cost aggregation

- VCN (NeurIPS 2019)
  - Multi-channel cost volume
  - Separable 4D conv
  - Soft Argmin

$$
\begin{aligned}
K(\mathbf{u},\mathbf{x}) * C(\mathbf{u},\mathbf{x}) &= \sum_{\mathbf{v},\mathbf{y}} K(\mathbf{v},\mathbf{y}) C(\mathbf{u}-\mathbf{v},\mathbf{x}-\mathbf{y}) && \textbf{[4D Convolution]} \\
&= \sum_{\mathbf{v},\mathbf{y}} \Big[K_{WTA}(\mathbf{v}) K_S(\mathbf{y}) \Big] C(\mathbf{u}-\mathbf{v},\mathbf{x}-\mathbf{y}) && \textbf{[Factorization]} \\
&= \sum_{\mathbf{v}} K_{WTA}(\mathbf{v}) \Big[\sum_{\mathbf{y}} K_S(\mathbf{y}) C(\mathbf{u}-\mathbf{v},\mathbf{x}-\mathbf{y}) \Big] && \textbf{[Separable Filtering]} \\
&= K_{WTA}(\mathbf{u}) * \Big[K_S(\mathbf{x}) * C(\mathbf{u},\mathbf{x}) \Big]
\end{aligned}
$$

Learning better cost aggregation

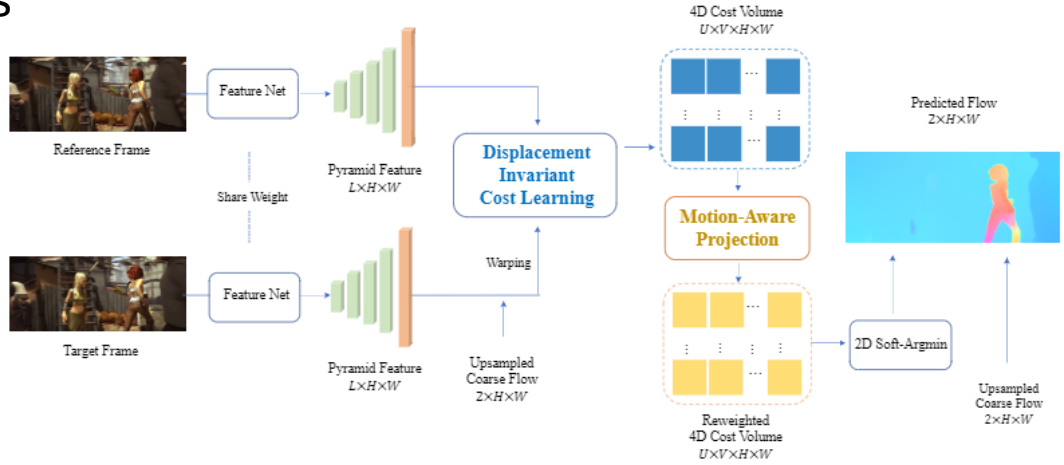


Taxonomy

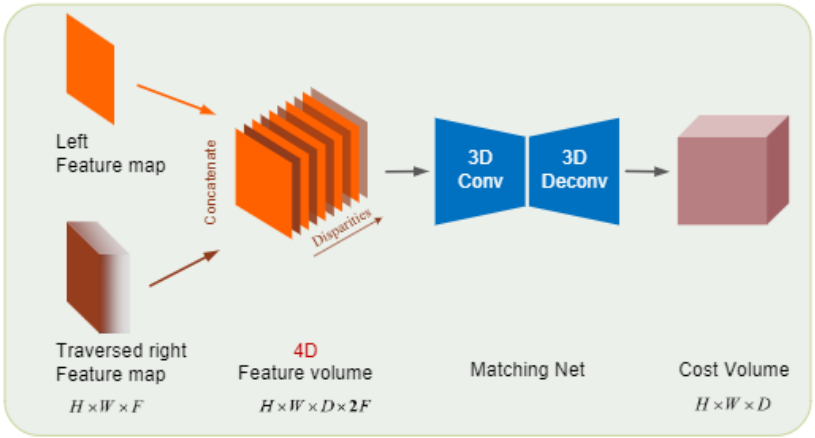
- Direct regression
- Learning better features
- Learning better cost aggregation
- Learning better cost metrics
- Learning better refinement
- Self-supervised deep optical flow

Learning better cost metrics

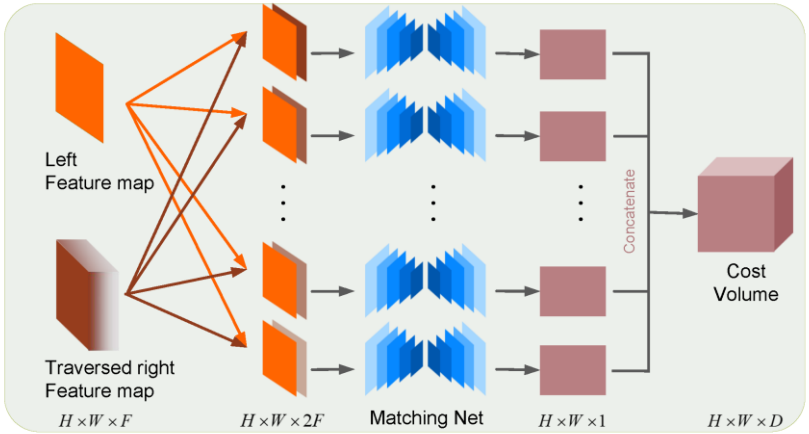
- DICLFlow
  - Learnable matching costs
  - No need 5D feature volume
  - No need 4D convolutions
  - 2D soft argmin



DICLFlow



Volumetric Methods

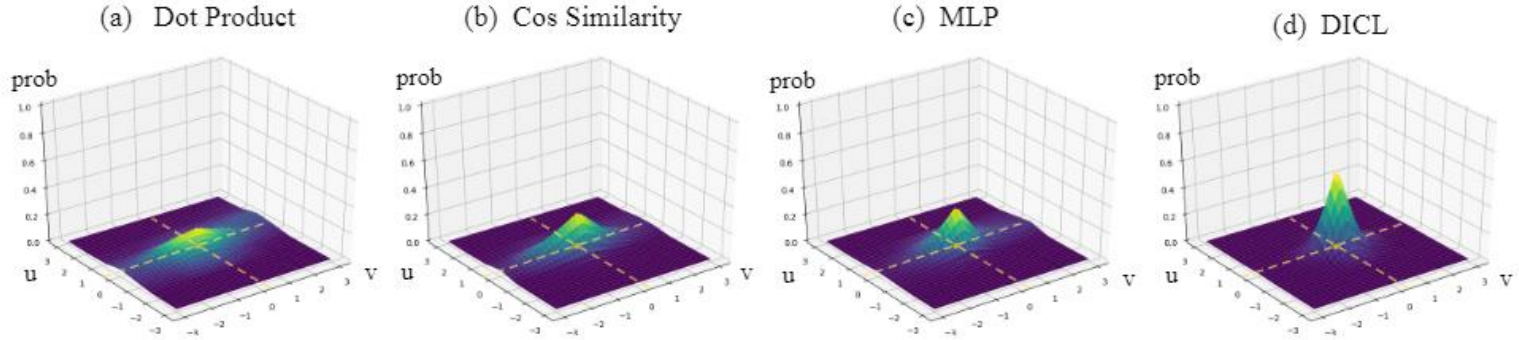


Displacement-Invariant Cost Learning

DICLFlow

Table 1: **Ablation study on cost computation metrics.** The models for 'Chair' were trained on the Chairs dataset. The models for 'K-15' and 'S' were trained on Things dataset.

Method	Chair	K-15 train		S-train (EPE)	
	EPE	EPE	Fl-all	Clean	Final
Dot Product	1.86	10.39	31.1	2.57	4.06
Cosine Simi	1.84	10.45	30.2	2.55	4.03
3-Layer MLP	1.76	9.83	28.9	2.45	3.98
Reduced DICL	1.72	9.77	28.3	2.42	3.99
DICL	1.33	8.78	23.8	2.11	3.85



DICLFlow

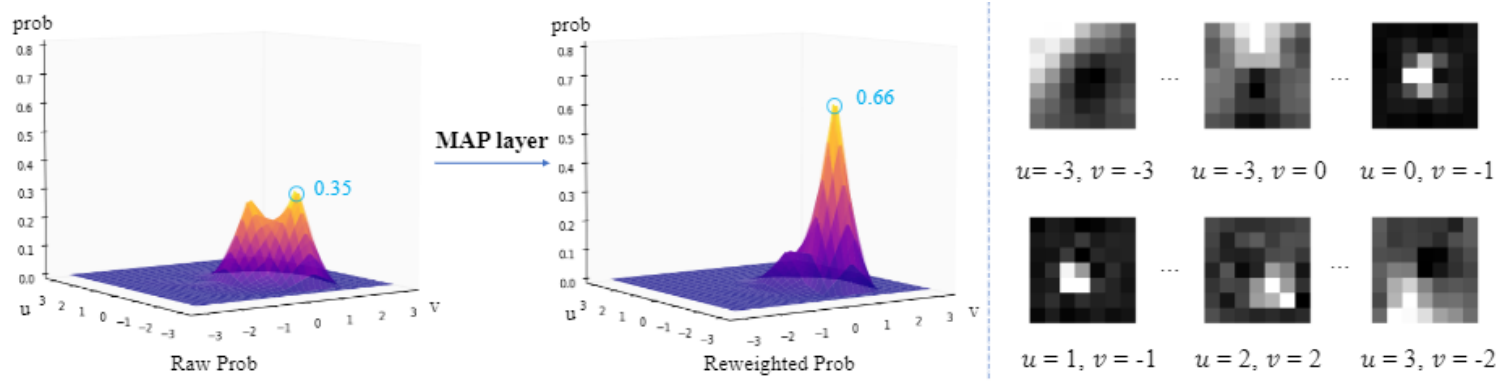


Figure 3: **Multi-Modal Effect and Visualization of the MAP Layer Kernels.** The left column compares an example pixel's displacement probability, before and after using MAP layer. The right column visualizes several kernels from a well-trained MAP layer, where white indicates a high value. The MAP layer has $U \times V$ kernels in total and each kernel maps the input costs to a specific u and v. The u and v below each kernel indicate the corresponding displacement hypothesis of its output cost. The qualitative analysis is provided in Section 4.3.

DICLFlow

Final | Clean

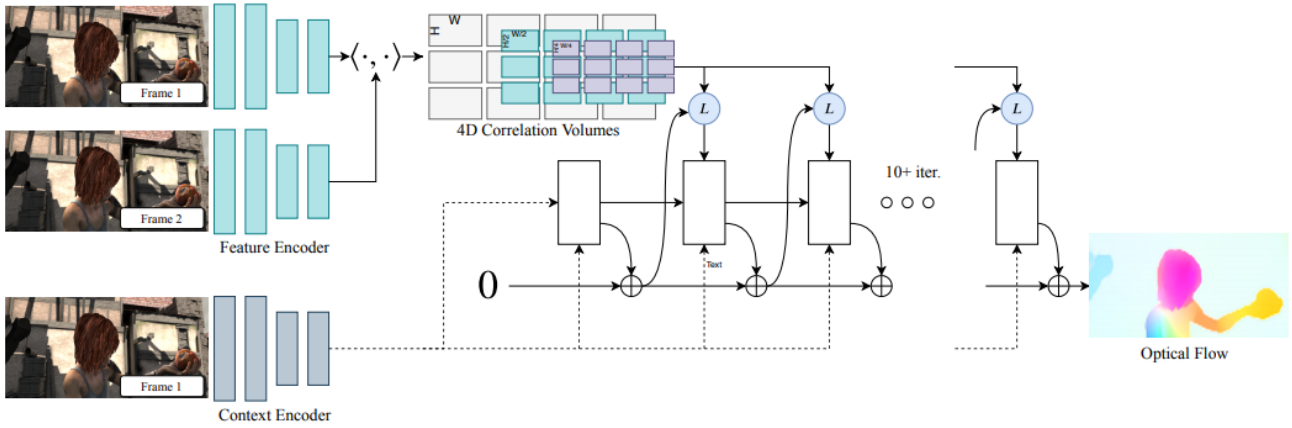
	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+	
GroundTruth [1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	Visualize Results
RAFT [2]	2.855	1.405	14.680	3.112	1.133	0.770	0.634	1.823	16.371	Visualize Results
DICL+ [3]	3.317	1.637	17.022	3.528	1.284	0.973	0.665	1.904	19.897	Visualize Results
RAFT-TF_RVC [4]	3.321	1.727	16.323	3.484	1.450	1.026	0.617	2.301	19.195	Visualize Results
ADLAB-PRFlow [5]	3.324	1.534	17.916	3.236	1.277	1.008	0.727	2.077	19.230	Visualize Results
RAFT+LCV [6]	3.365	1.589	17.848	3.801	1.244	0.911	0.792	2.236	18.858	Visualize Results
MRFP [7]	3.427	1.570	18.566	3.609	1.151	0.929	0.717	2.077	20.140	Visualize Results
PRAFlow_RVC [8]	3.559	1.629	19.292	2.851	1.322	1.118	0.545	1.799	22.770	Visualize Results

Taxonomy

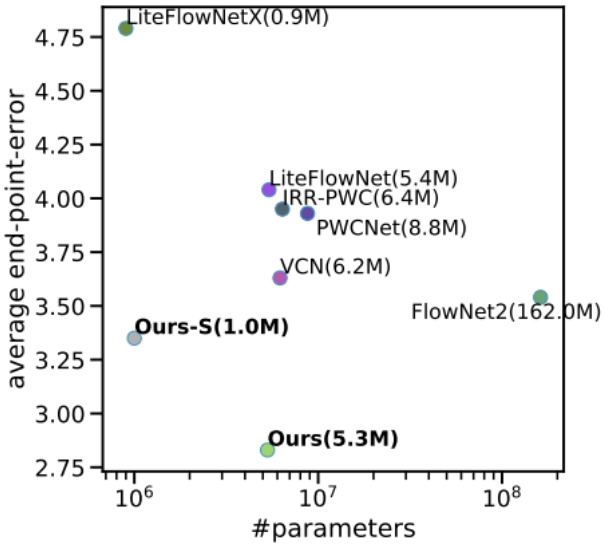
- Direct regression
- Learning better features
- Learning better cost aggregation
- Learning better cost metrics
- Learning better refinement
- Self-supervised deep optical flow

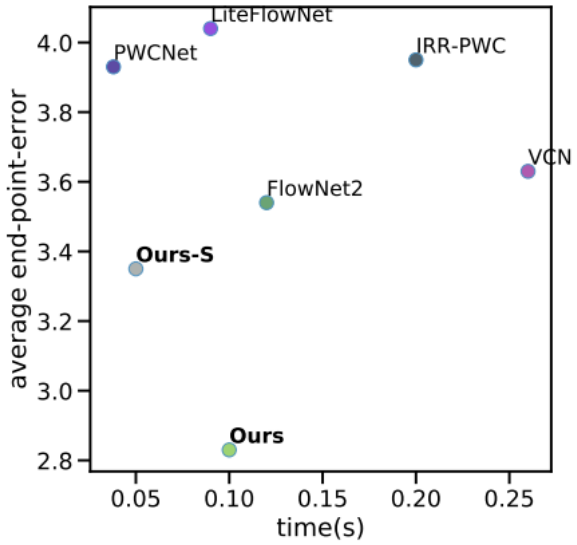
Learning better refinement

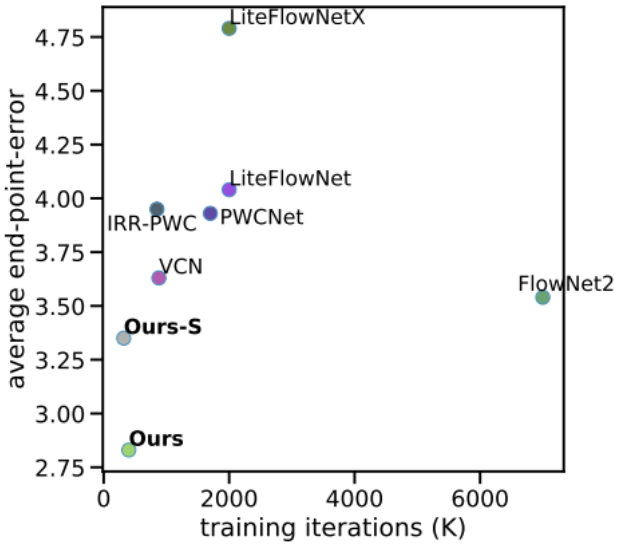
- RAFT (ECCV 2020)
  - All-Pairs: cost volume with a size of HxWxHxW
  - Recurrent refinement



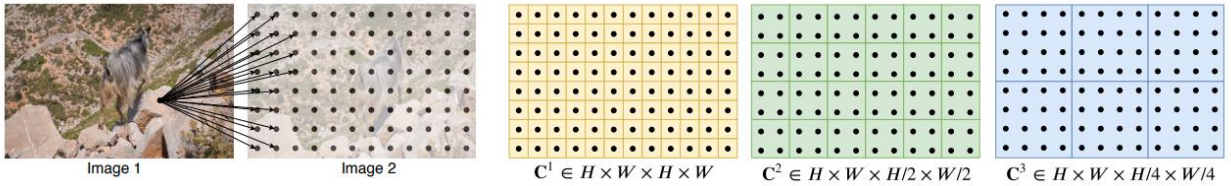
RAFT


LiteFlowNetX(0.9M)
LiteFlowNet(5.4M)
IRR-PWC(6.4M)
PWCNet(8.8M)
VCN(6.2M)
FlowNet2(162.0M)
Ours-S(1.0M)
Ours(5.3M)
average end-point-error
#parameters


LiteFlowNet
PWCNet
IRR-PWC
VCN
FlowNet2
Ours-S
Ours
average end-point-error
time(s)


LiteFlowNetX
LiteFlowNet
IRR-PWC
PWCNet
VCN
FlowNet2
Ours-S
Ours
average end-point-error
training iterations (K)

RAFT



- Key to success
  - Single scale features
  - Lookup rather than warping

$$F(\mathbf{p}_1) = 0, \quad F(\mathbf{p}_2) = \mathbf{p}_1 - \mathbf{p}_2, \tag{2}$$

we have

$$\begin{aligned} \tilde{J}(\mathbf{p}_1) &= J(\mathbf{p}_1 + F(\mathbf{p}_1)) && (3)\\ &= J(\mathbf{p}_1 + 0) = J(\mathbf{p}_1), && (4)\\ \tilde{J}(\mathbf{p}_2) &= J(\mathbf{p}_2 + F(\mathbf{p}_2)) && (5)\\ &= J(\mathbf{p}_2 + \mathbf{p}_1 - \mathbf{p}_2) = J(\mathbf{p}_1). && (6) \end{aligned}$$

Therefore $\tilde{J}(\mathbf{p}_1) = \tilde{J}(\mathbf{p}_2) = J(\mathbf{p}_1)$. Since the value of $J(\mathbf{p}_1)$ is unique in image J but not unique in $\tilde{J}$, a duplicate is created on the warped second image $\tilde{J}$.

(a) First image (b) Second image

(c) Ground truth optical flow (d) Warped second image

Taxonomy

- Direct regression
- Learning better features
- Learning better cost aggregation
- Learning better cost metrics
- Learning better refinement
- Self-supervised deep optical flow

Self-supervised deep optical flow

- EpiFlow (CVPR 2019)
  - Leveraging epipolar constraints in stationary and dynamic scenes
    - Explicit fundamental matrix constraint;
    - Low-rank constraint;
    - Union-of-subspaces constraint.
- SelFlow (CVPR 2019)
  - Synthesize occlusions to learn occlusion handling

Deep Epipolar Flow

- Epipolar Constraints

$$\boldsymbol{x}_i'^T \boldsymbol{F} \boldsymbol{x}_i = 0.$$

Where $\boldsymbol{x_i'} = (x_i', y_i', 1)^T$ and $\boldsymbol{x_i} = (x_i, y_i, 1)^T$ are correspondences between two frames.

Deep Epipolar Flow

- Sampson Distance
  - first order approximation of Gold Standard method

$$L_F = \sum_{i}^{N} \frac{(\boldsymbol{x}_i'^T \boldsymbol{F} \boldsymbol{x}_i)^2}{(\boldsymbol{F}\boldsymbol{x}_i)_1^2 + (\boldsymbol{F}\boldsymbol{x}_i)_2^2 + (\boldsymbol{F}^T\boldsymbol{x}_i')_1^2 + (\boldsymbol{F}^T\boldsymbol{x}_i')_2^2}$$

Chicken-and-egg problem:

1. Estimating a fundamental matrix $\boldsymbol{F}$ from an estimated flow.
2. Updating the flow to comply with the $\boldsymbol{F}$.

Deep Epipolar Flow

- Sampson Distance
  - For multi-motion scenes:
    - Update flow estimation;
    - Estimate $\boldsymbol{F}^m$ for each rigid motion given current motion segmentation;
    - Update motion segmentation based on the nearest $\boldsymbol{F}^m$.

Deep Epipolar Flow

- Low-rank Constraint
  - Soft constraint without explicitly computing $\boldsymbol{F}$;
  - Static scene.

$$\boldsymbol{x}_i'^T \boldsymbol{F} \boldsymbol{x}_i = 0.$$

$$\boldsymbol{f}^\top vec\left(\boldsymbol{x}_i' \boldsymbol{x}_i^T\right) = 0$$

Where $\boldsymbol{f} \in \mathbb{R}^9$ is the vectorized $\boldsymbol{F}$, and

$vec\left(\boldsymbol{x}_i' \boldsymbol{x}_i^T\right) = (x_i x_i', x_i y_i', x_i, y_i x_i', y_i y_i', y_i, x_i', y_i', 1)^T$.

Epipolar subspace

Deep Epipolar Flow

- Low-rank Constraint

$$\boldsymbol{h}_i = vec\left(\boldsymbol{x}_i'\boldsymbol{x}_i^T\right),$$

$$\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N],$$

$$L_{lowrank} = rank(\boldsymbol{H}),$$

Its convex surrogate:

$$L^*_{lowrank} = \|\boldsymbol{H}\|_*,$$

can be computed by SVD.

Deep Epipolar Flow

- Union-of-Subspaces Constraint
  - Applicable for dynamic scenes.

$$\boldsymbol{h}_i = vec\left(\boldsymbol{x}_i'\boldsymbol{x}_i^T\right) = (x_i x_i', x_i y_i', x_i, y_i x_i', y_i y_i', y_i, x_i', y_i', 1)^T$$

$\boldsymbol{h}_i$ lie in a union of subspace. Therefore, we have

$$\min_{\boldsymbol{C}} \frac{1}{2}\|\boldsymbol{C}\|_F^2 \quad \text{s.t. } \boldsymbol{H} = \boldsymbol{H}\boldsymbol{C}$$

$\boldsymbol{C}$ is the subspace self-expression coefficient

Deep Epipolar Flow

- Union-of-Subspaces Constraint
  - Dealing with noises

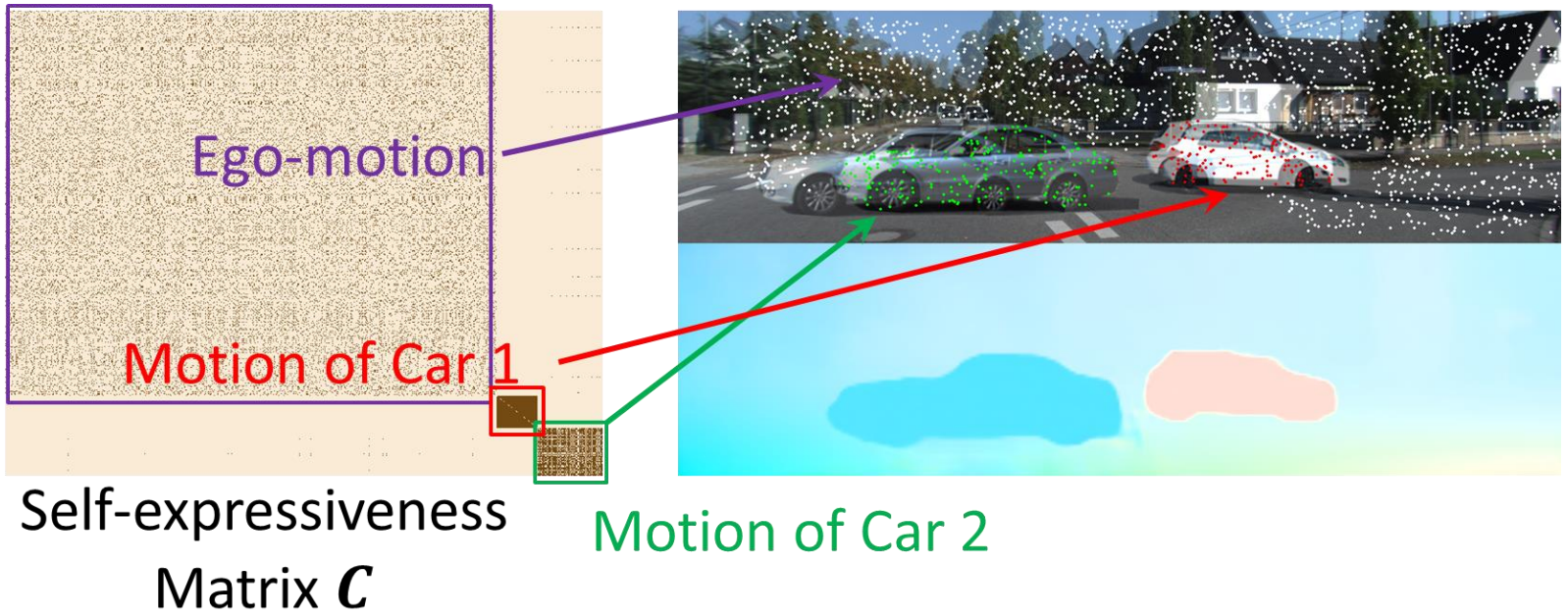
$$L_{subspace} = \frac{1}{2}\|\boldsymbol{C}\|_F^2 + \frac{\lambda}{2}\|\boldsymbol{HC} - \boldsymbol{H}\|_F^2$$

$$\boldsymbol{C}^* = (\boldsymbol{I} + \lambda \boldsymbol{H}^T\boldsymbol{H})^{-1}\lambda \boldsymbol{H}^T\boldsymbol{H}$$

$$L_{subspace} = \frac{1}{2}\|(\boldsymbol{I} + \lambda \boldsymbol{H}^T\boldsymbol{H})^{-1}\lambda \boldsymbol{H}^T\boldsymbol{H}\ \|_F^2 + \frac{\lambda}{2}\|\boldsymbol{H}(\boldsymbol{I} + \lambda \boldsymbol{H}^T\boldsymbol{H})^{-1}\lambda \boldsymbol{H}^T\boldsymbol{H}\ - \boldsymbol{H}\|_F^2$$

Deep Epipolar Flow

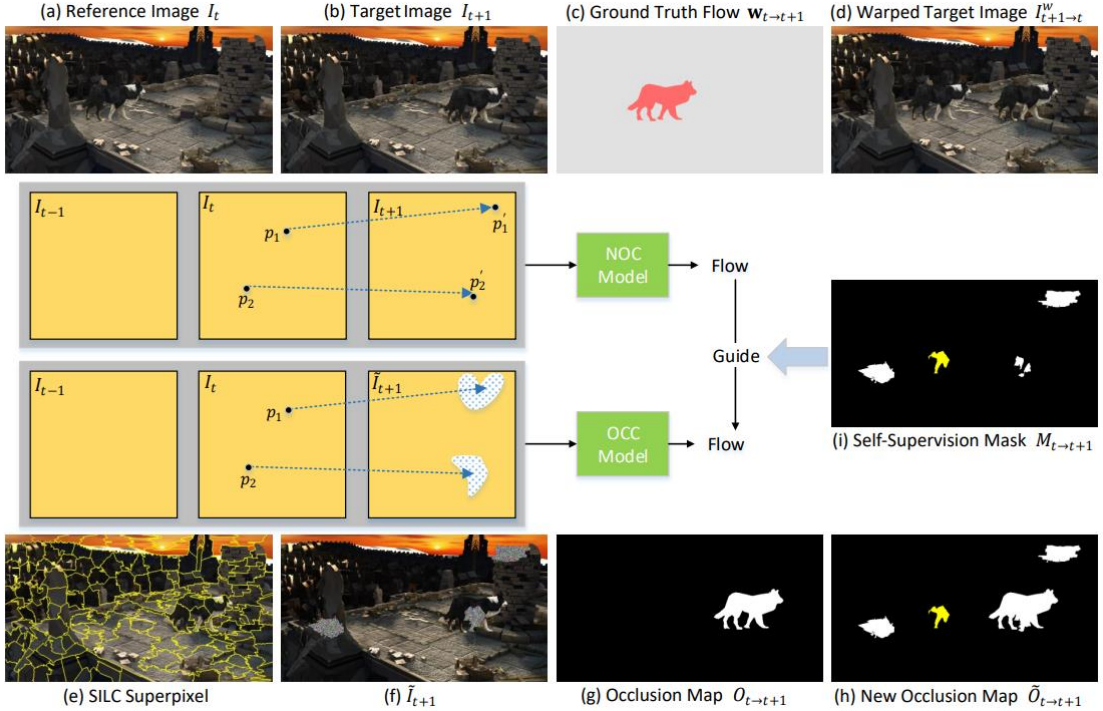
- Union-of-Subspaces Constraint



Unsupervised deep optical flow

- EpiFlow (CVPR 2019)
  - Leveraging epipolar constraints in stationary and dynamic scenes
    - Explicit fundamental matrix constraint;
    - Low-rank constraint;
    - Union-of-subspaces constraint.
- SelFlow (CVPR 2019)
  - Synthesize occlusions to learn occlusion handling

SelFlow


(a) Reference Image I_t
(b) Target Image I_{t+1}
(c) Ground Truth Flow $\mathbf{w}_{t\to t+1}$
(d) Warped Target Image $I^w_{t+1\to t}$
I_{t-1}
I_t
I_{t+1}
p_1
p_2
p_1'
p_2'
NOC Model
Flow
Guide
OCC Model
Flow
$\tilde{I}_{t+1}$
(i) Self-Supervision Mask $M_{t\to t+1}$
(e) SILC Superpixel
(f) $\tilde{I}_{t+1}$
(g) Occlusion Map $O_{t\to t+1}$
(h) New Occlusion Map $\tilde{O}_{t\to t+1}$

Robust deep optical flow

- Attacking optical flow (ICCV 2019)

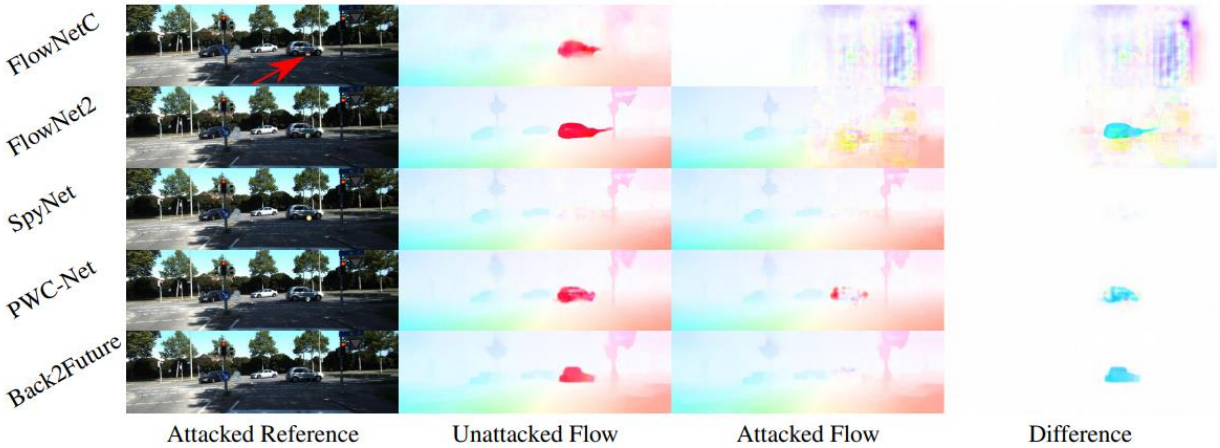


Figure 3. **White-box Attacks.** 25x25 patch attacks on all networks.

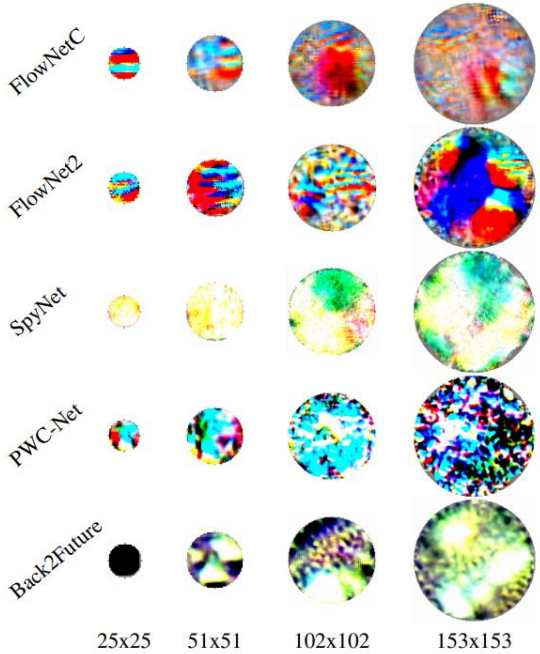
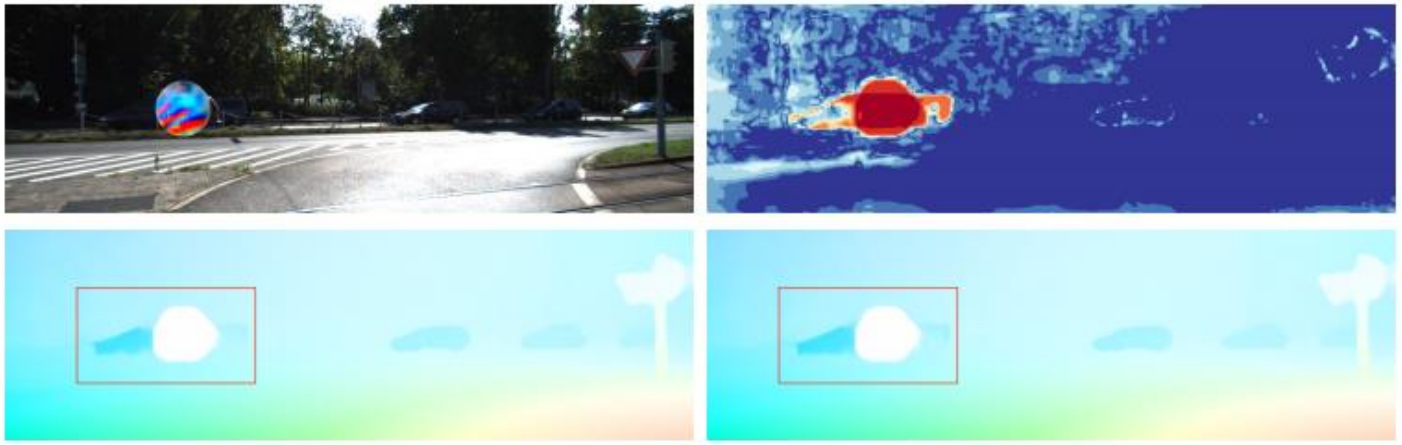


Figure 2. **Adversarial Patches**. Obtained for different optical flow networks. The size is enlarged for visualization purposes.

Robust deep optical flow

- VCN



Robust deep optical flow

- DICLFlow

Attacked Reference Frame

Unattacked Flow

Attacked Flow

Network	Unattacked EPE	25x25 EPE	25x25 Diff	51x51 EPE	51x51 Diff	102x102 EPE	102x102 Diff	153x153 EPE	153x153 Diff
FlowNetC [3]	14.56	29.07	+14.51	40.27	+25.51	82.41	+67.85	95.32	+80.76
FlowNet2 [4]	11.90	17.04	+5.14	24.42	+12.52	38.57	+26.67	59.58	+47.68
SpyNet [8]	20.26	20.59	+0.33	21.00	+0.74	21.22	+0.96	21.00	+0.74
PWC-Net [10]	11.03	11.37	+0.34	11.50	+0.47	11.86	+0.83	12.52	+1.49
Back2Future [5]	17.49	18.04	+0.55	18.24	+0.75	18.73	+1.24	18.43	+0.94
Ours	8.98	9.17	+0.19	9.30	+0.32	9.52	+0.54	9.61	+0.63

Robust deep optical flow

- Key to robust
  - Spatial pyramid, i.e., coarse to fine
  - Task-specific knowledge
  - Soft argmin

Limit the free-space of a network

Future directions

- Robustness
- Accuracy
- Combine with other tasks, i.e., action recognition
- Application
  - Time to Collision