Lecture 18

Motion estimation

Deqing Sun (Google)

Credits

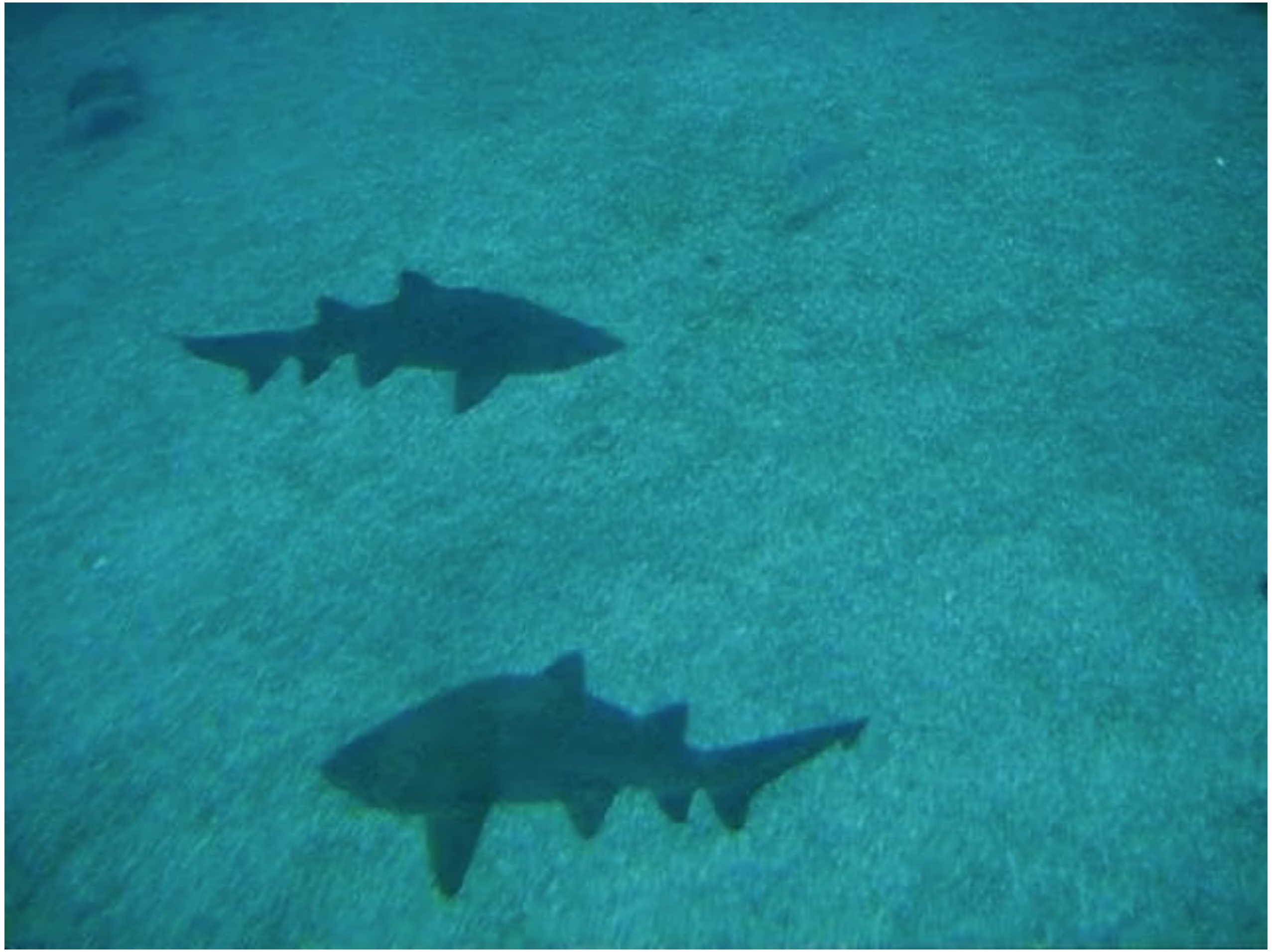
Some slides, images, and videos from

- Dr. Ce Liu@Microsoft
- Dr. Huaizu Jiang@Northeastern
- Dr. David Fouhey@UMich
- Dr. Justin Johnson@UMich
- Dr. Svetlana Lazebnik@UIUC
- Dr. Shree K. Nayar@Columbia
- Dr. Jia Deng@Princeton
- Dr. Ming-Hsuan Yang@UC Merced
- Dr. Rick Szeliski’s book
- Book by Antonio, Phillip, and Bill

Suggestions from Dr. Bill Freeman, Dr. Rick Szeliski, Dr. Noah Snavely and Dr. Junhwa Hur

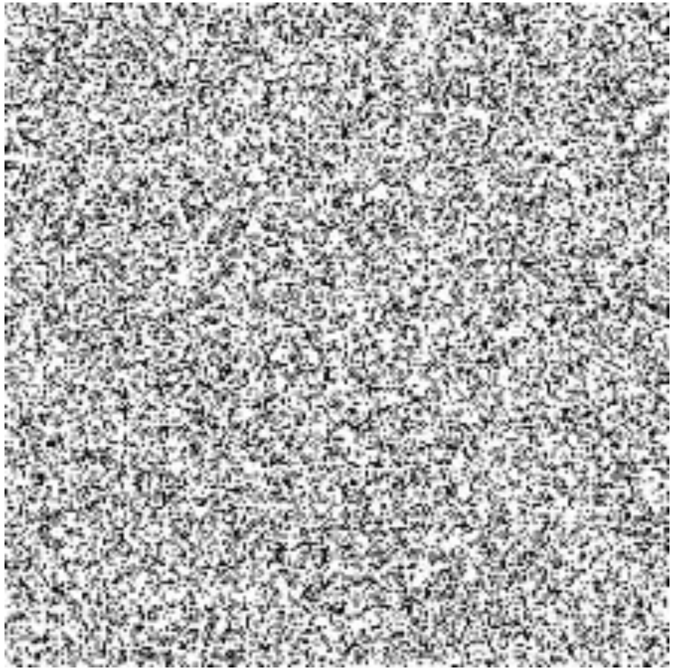
We live in a dynamic world

Perceiving, understanding and predicting motion is an important part of our daily lives



Motion and perceptual organization

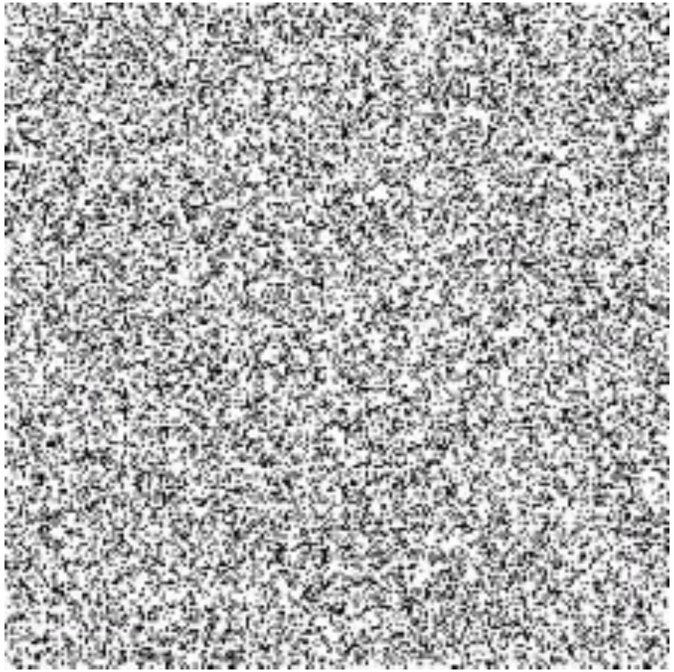
Sometimes motion is the only cue



Slide Credit: S. Lazebnik, but idea of random dot sterogram is due to B. Julesz

Motion and perceptual organization

Sometimes motion is the only cue



Slide Credit: S. Lazebnik, but idea of random dot sterogram is due to B. Julesz

Motion and perceptual organization

Even impoverished data can create a strong percept

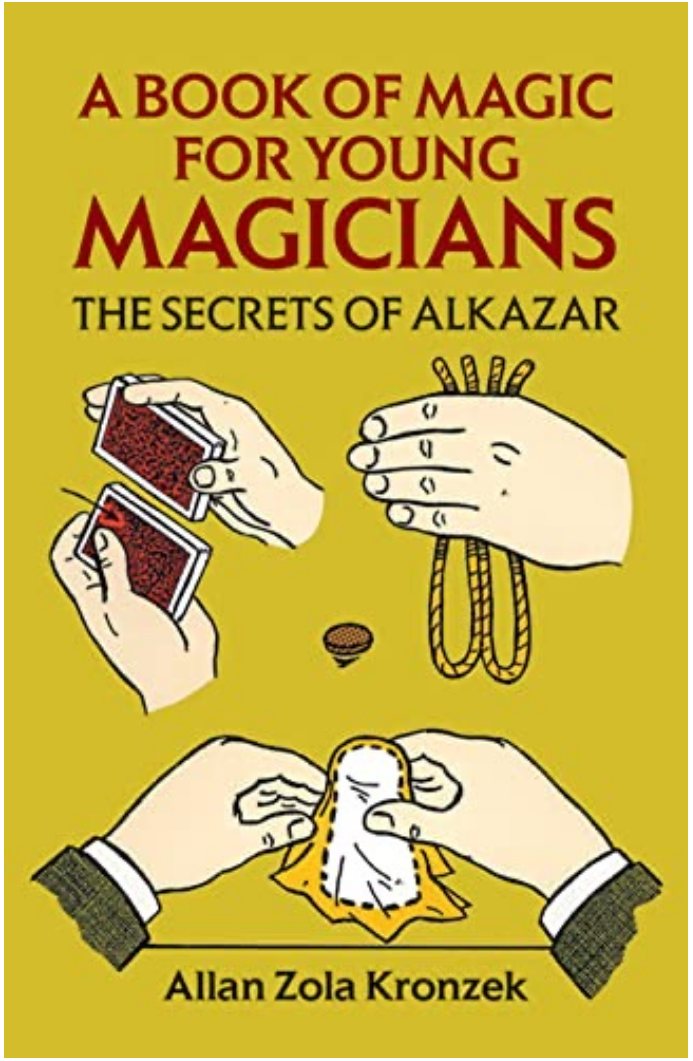
Slide Credit: S. Lazebnik

Motion and perceptual organization

Even impoverished data can create a strong percept

Slide Credit: S. Lazebnik

We pay attention to motion



Alkazar's principles of misdirection

The key to misdirection lies in learning to control **attention**.

Principle 1

The audience will pay attention to what **moves**. ...

What doesn't **move** ... doesn't attract attention.

…

Content

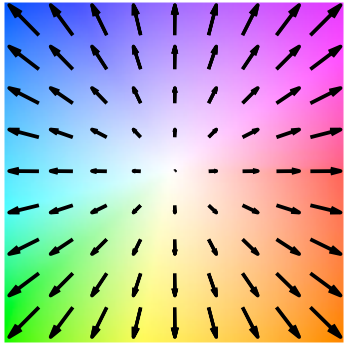
- Classical approach
- Deep learning-based approach
- Applications: What is motion for?

Classical approach

Optical flow: 2D motion of every pixel

Input [Liu *et al.* CVPR'08]

Optical flow (2D motion vector)

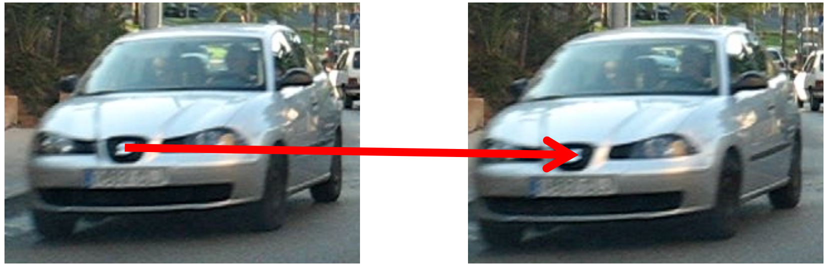


Color key
[Baker *et al.* IJCV'11]

Fundamental assumption: Brightness constancy

[Horn & Schunck AI'81]

$$I_t(\mathbf{p}) \approx I_{t+1}(\mathbf{p} + \mathbf{w_p})$$



First image (t) Second image (t+1)

Matching-based motion estimation

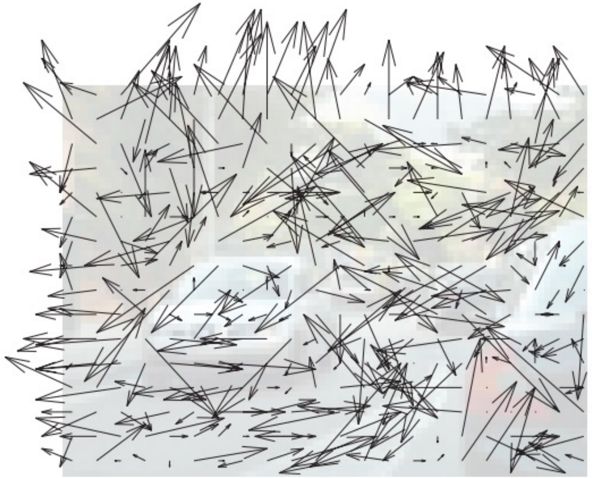
Similarity between a pixel in image 1 with pixels in image 2

$$\min_{\mathbf{w_p}} \left(I_t(\mathbf{p}) - I_{t+1}(\mathbf{p} + \mathbf{w_p}) \right)^2$$

Image 1

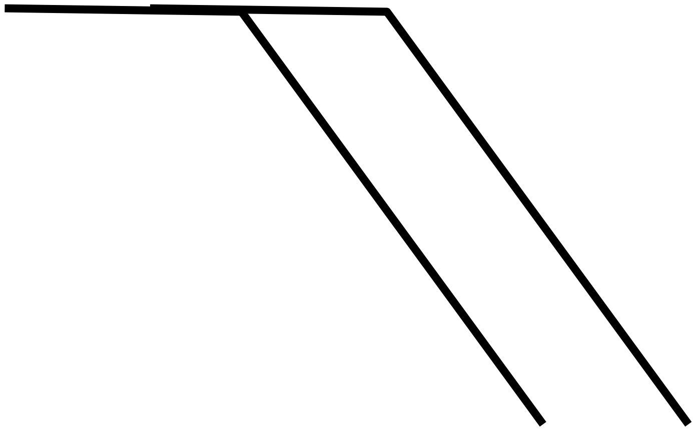
Image 2

Comparing pixel colors



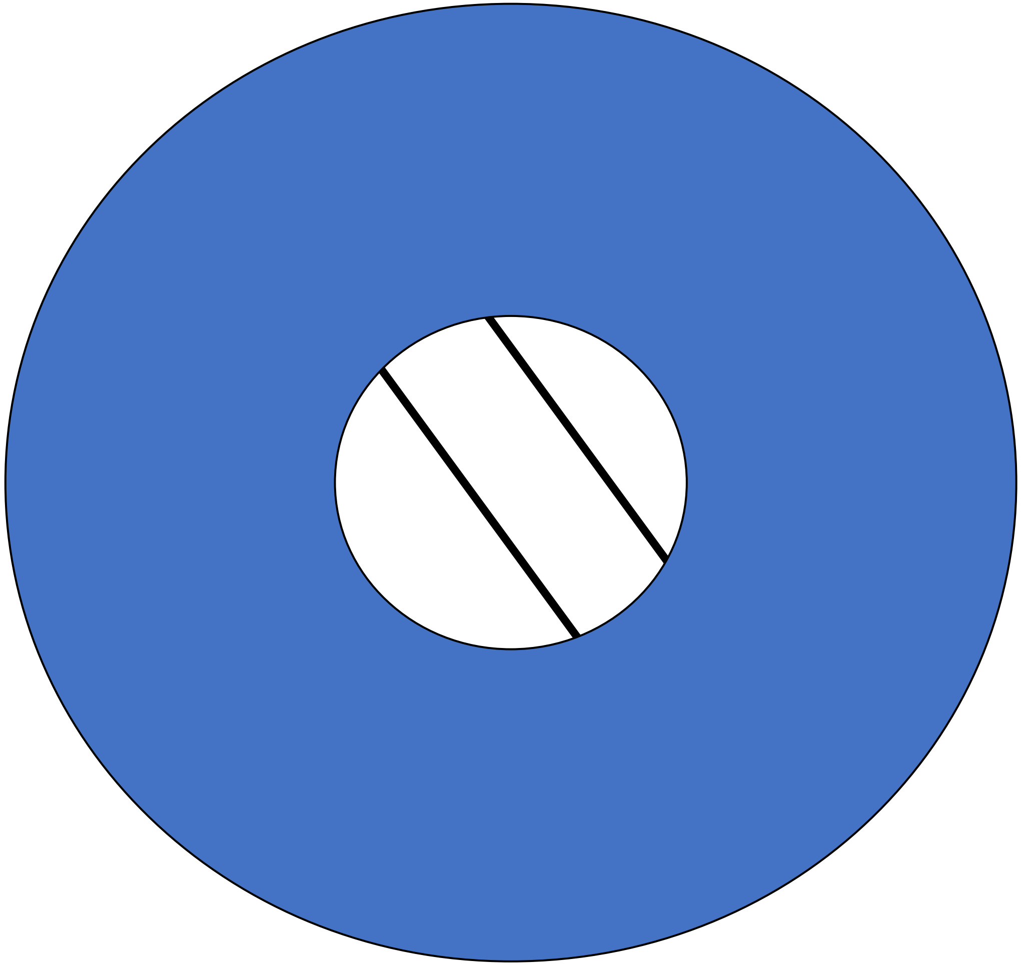
1x1

Aperture problem



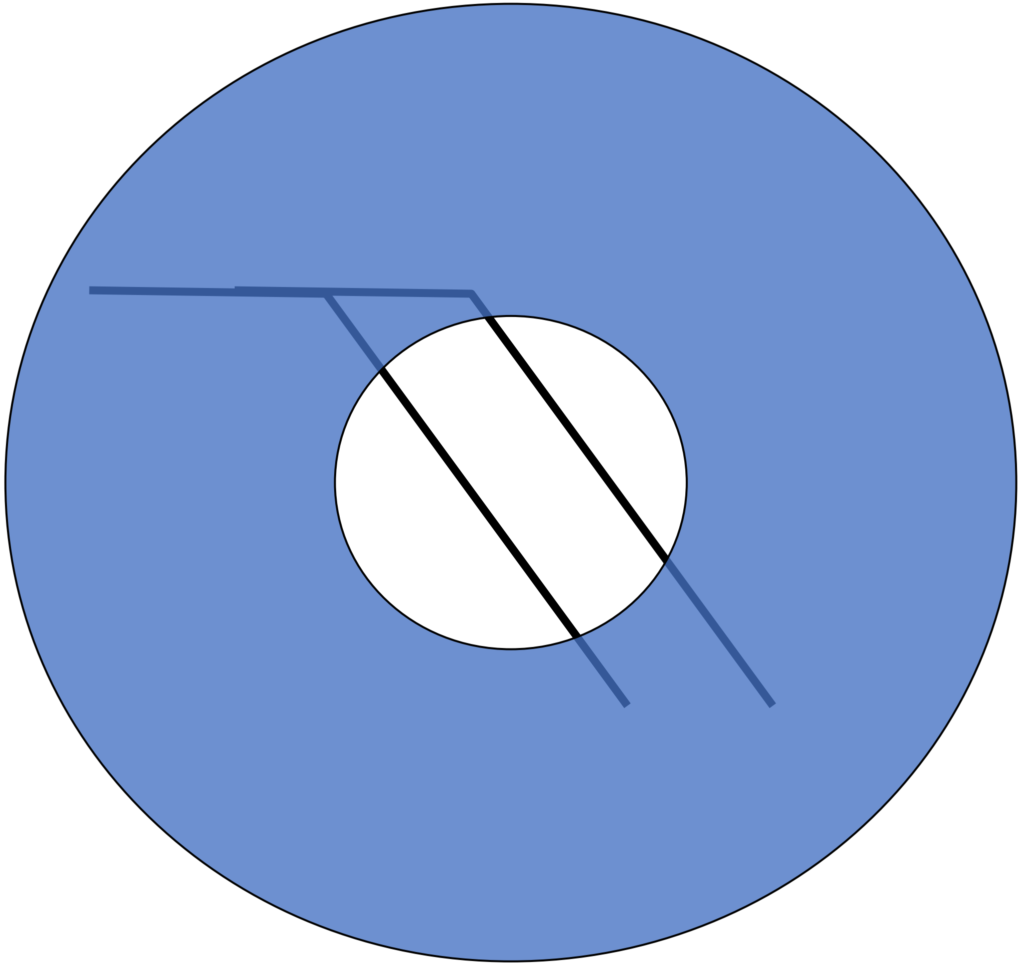
Slide Credit: S.

Aperture problem



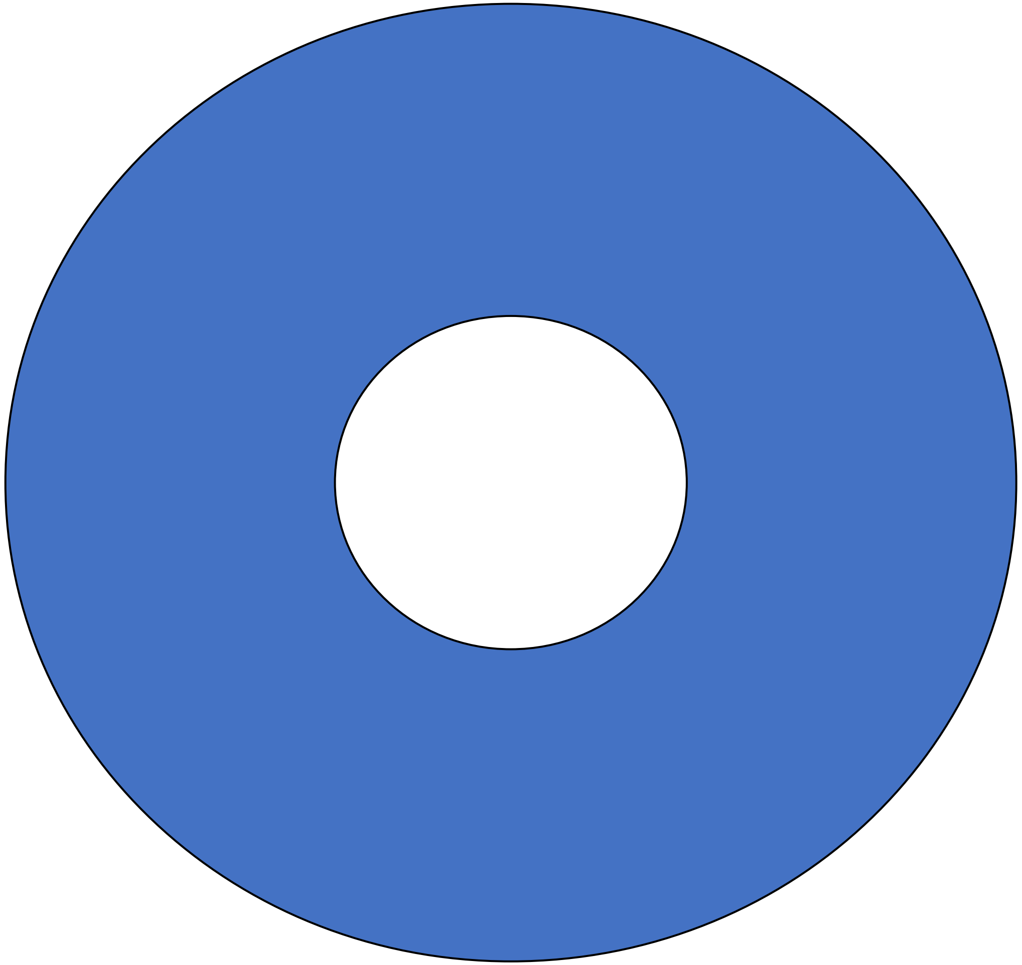
Slide Credit: S. Lazebnik

Aperture problem



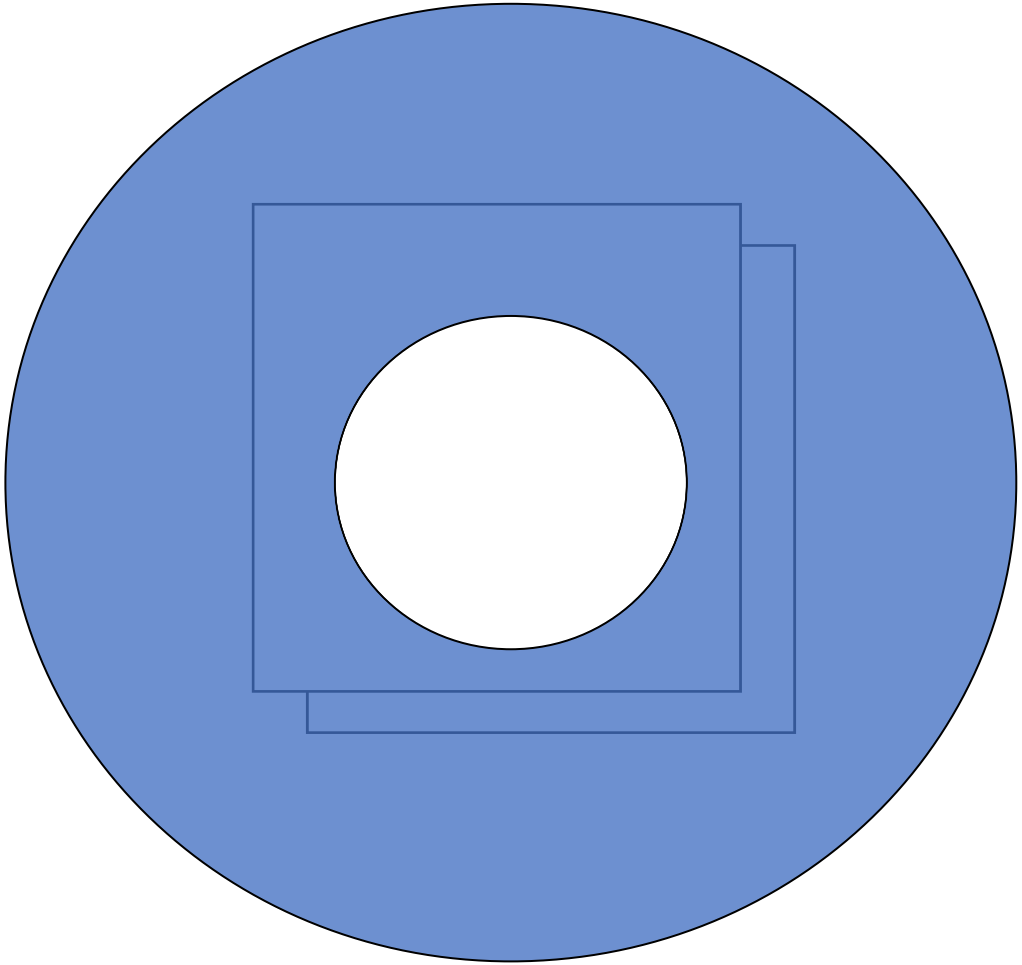
Slide Credit: S. Lazebnik

Other invisible flow



Slide credit: D Fouhey & J Johnson

Other invisible flow



Slide credit: D Fouhey & J Johnson

What do you perceive?

Actual motion

Matching-based motion estimation

Similarity between a pixel in image 1 with pixels in image 2

$$\min_{\mathbf{w_p}} \left(I_t(\mathbf{p}) - I_{t+1}(\mathbf{p} + \mathbf{w_p}) \right)^2$$

Image 1

Image 2

Solving ambiguities: Lucas-Kanade

Similarity between a patch in image 1 with patches in image 2

- Pixels in a patch share the same (parametric) motion

$$\min_{\mathbf{w_p}} \sum_{\mathbf{q} \in \boldsymbol{N_p}} \left(I_t(\mathbf{q}) - I_{t+1}(\mathbf{q} + \mathbf{w_p}) \right)^2$$

Image 1

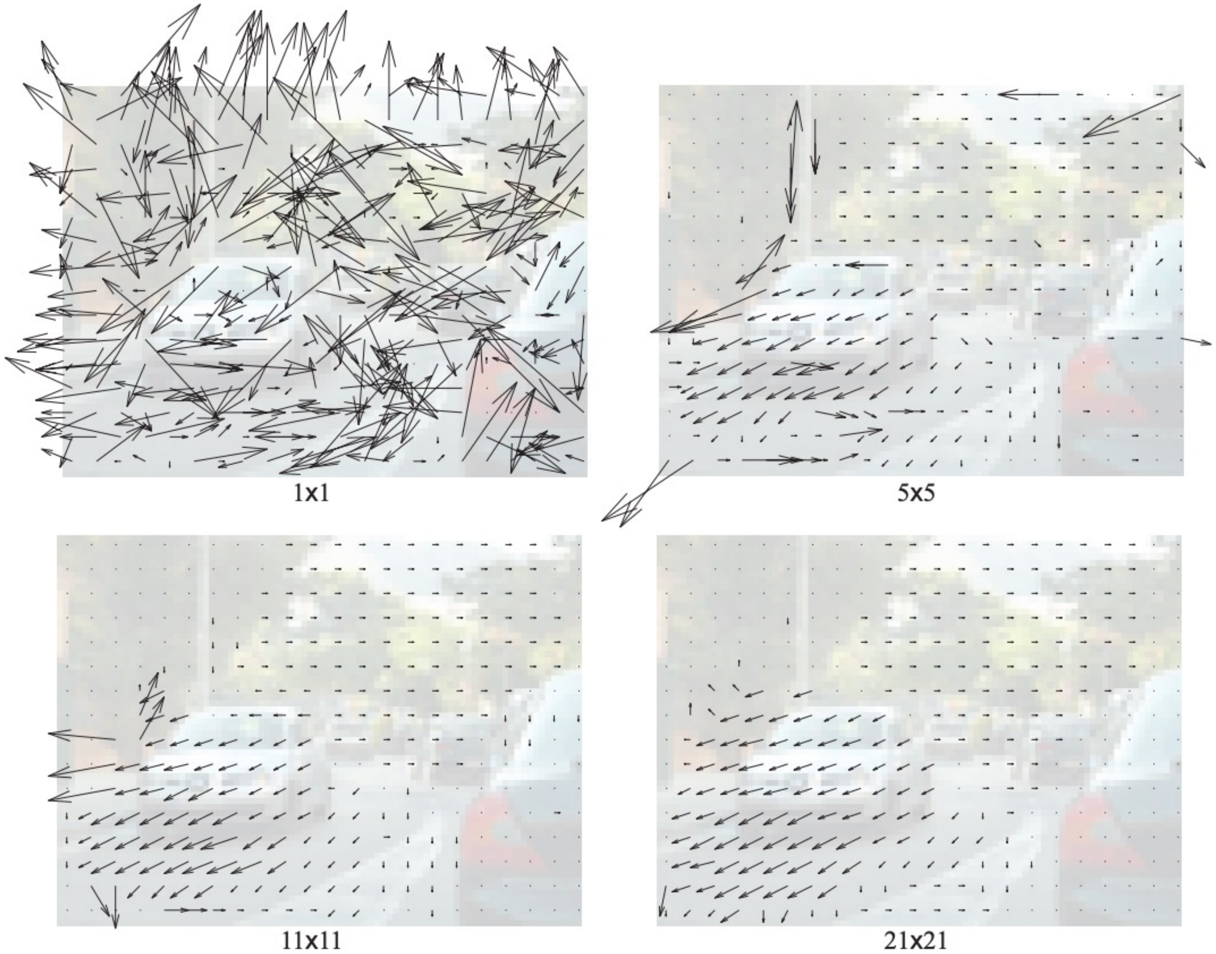
Image 2

Similarity between patches (cost volume)

Image 1

Cost volume (darker, more similar)

Effect of patch size



Issue: Brute force is too expensive

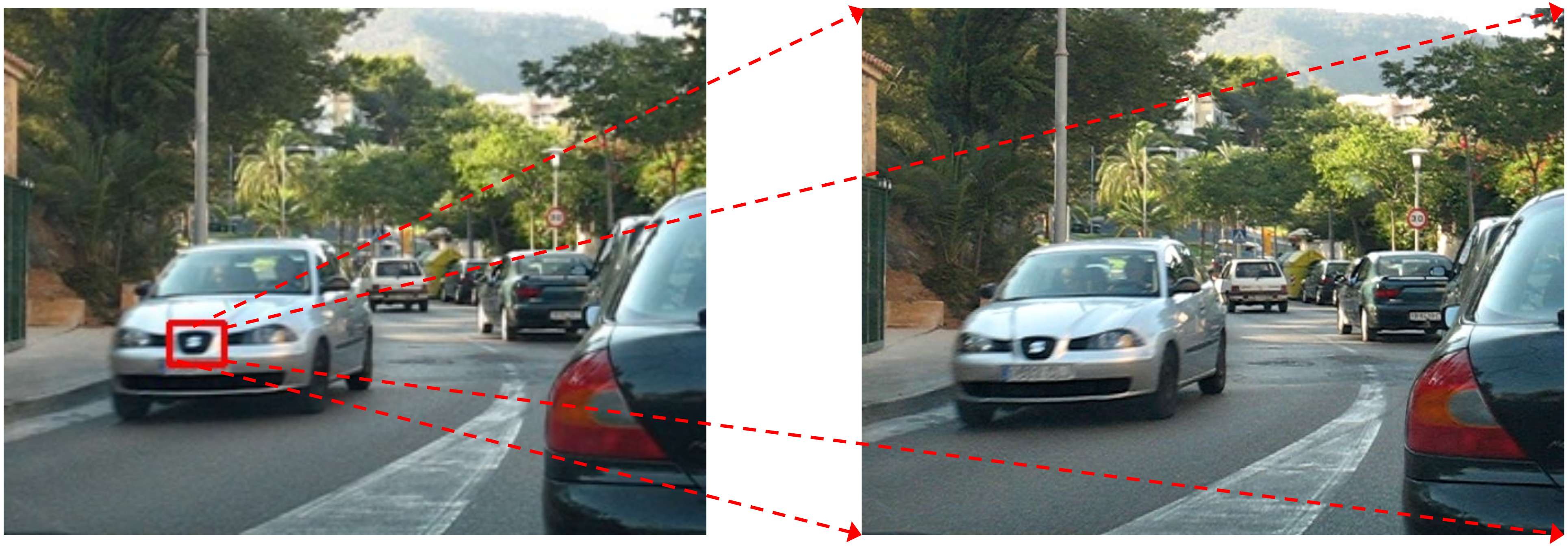
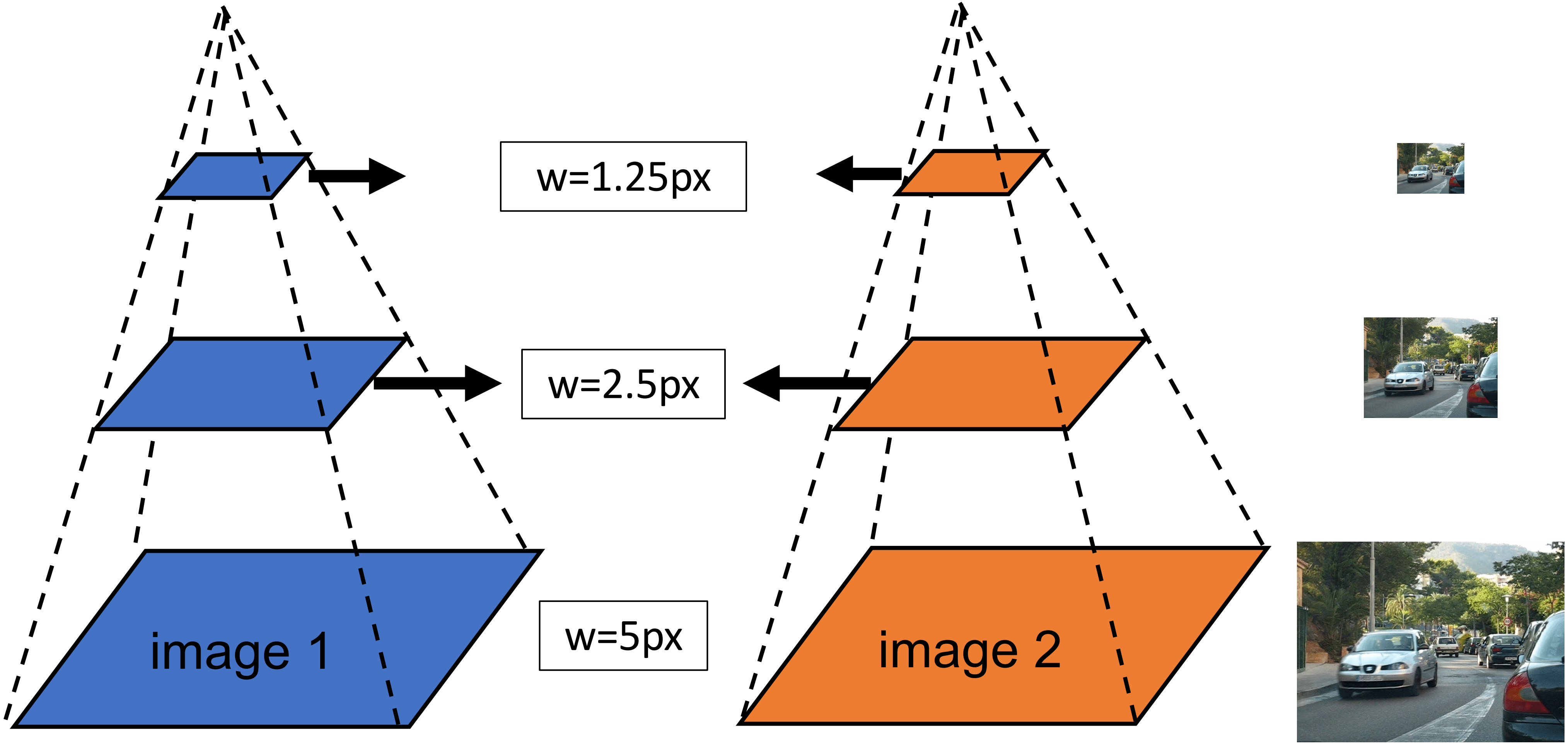


Image 1

Image 2

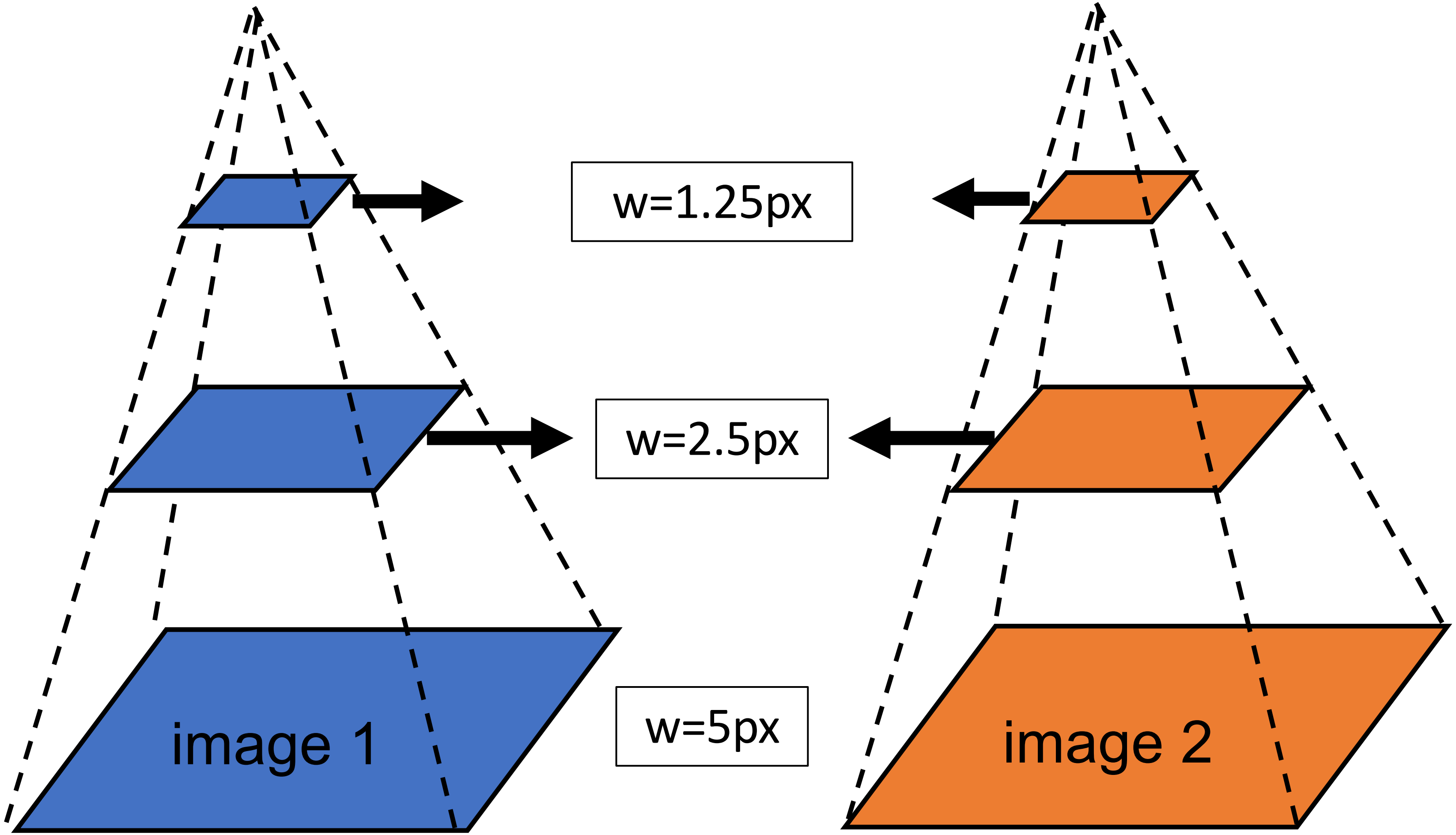
Coarse-to-fine iterative estimation



Start from top or bottom?

Slide credit: S. Lazebnik

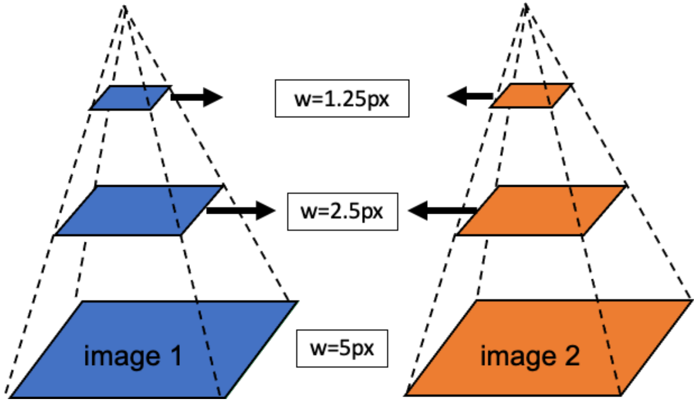
Coarse-to-fine iterative estimation



How to use estimates from the upper level?

Slide credit: S. Lazebnik

Coarse-to-fine iterative estimation



Current estimate

$$I_{t+1}(\mathbf{q} + \mathbf{w}_\mathbf{p}) = I_{t+1}(\mathbf{q} + \mathbf{w}_\mathbf{p}^k + \boldsymbol{\delta}\mathbf{w}_p)$$

Small increment

Warped image

$$I_\mathrm{w}(\mathbf{q}) = I_{t+1}(\mathbf{q} + \mathbf{w}_\mathbf{p}^k)$$

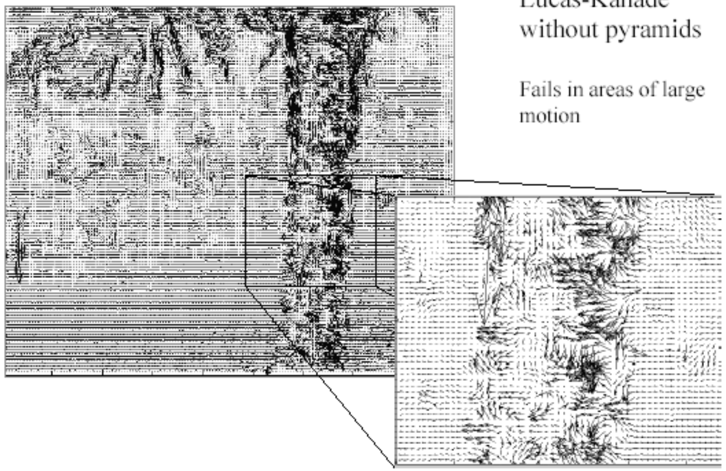
Warping operation

$$I_{\mathrm{w}}(\mathbf{q}) = I_{t+1}\left(\mathbf{q} + \mathbf{w}_{\mathbf{p}}^{\boldsymbol{k}}\right)$$

Input images *t* and *t*+1

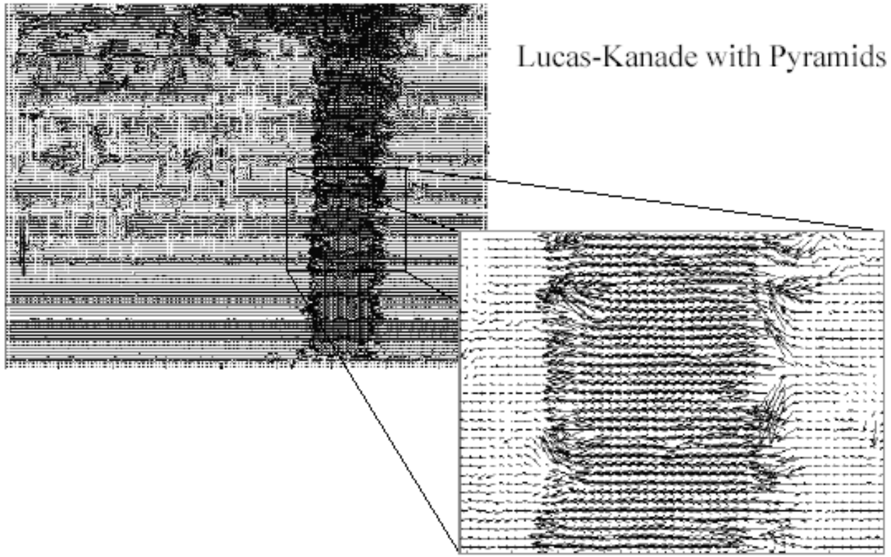
Image *t* and warped image

Optical Flow Results



Lucas-Kanade
without pyramids

Fails in areas of large
motion



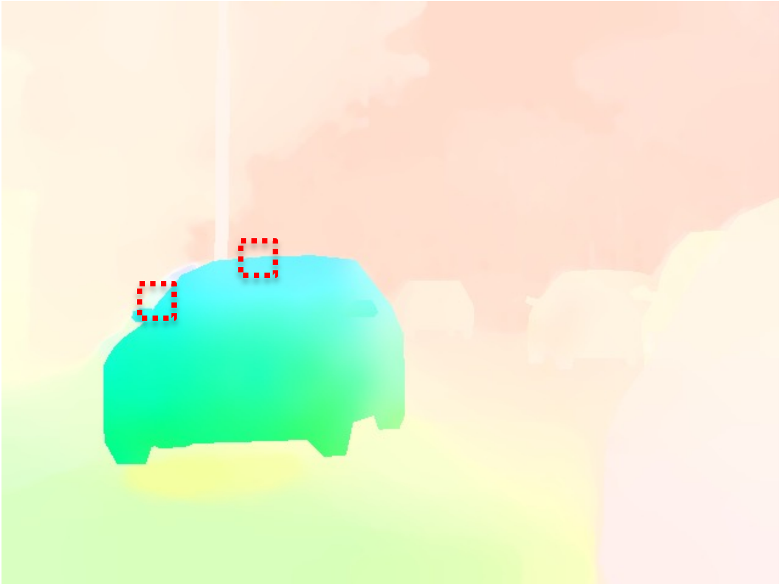
Lucas-Kanade with Pyramids

Slide credit: S. Lazebnik

Issue: Motion boundaries

Pixels in a patch share the same (parametric) motion

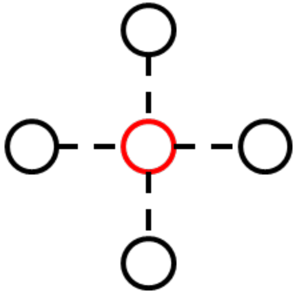
$$\min_{\mathbf{w_p}} \sum_{\mathbf{q} \in N_p} \left(I_t(\mathbf{q}) - I_{t+1}(\mathbf{q} + \mathbf{w_p}) \right)^2$$



Solving ambiguities: Horn & Schunck

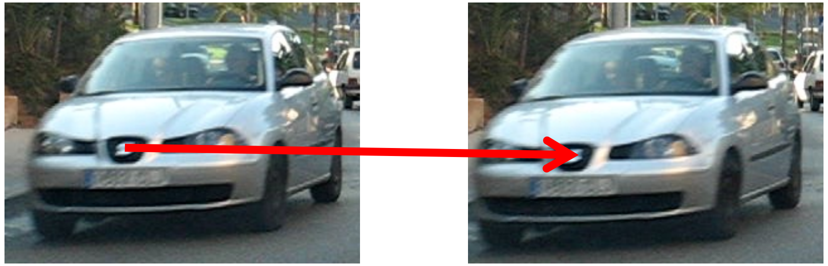
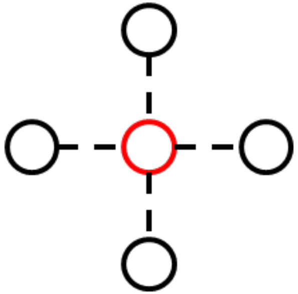
Smoothness: neighboring pixels have similar motion

$$\mathbf{w}_{\mathbf{p}} \approx \mathbf{w}_{\mathbf{q}},\ \mathbf{q} \in N_{\mathbf{p}}$$



Optimization/energy minimization

[Horn & Schunck AI'81]

() + 

Constancy (data) term

Smoothness (prior) term

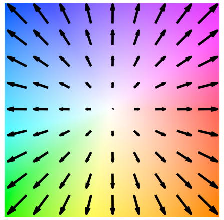
$$E(\mathbf{w}) = \sum_{\mathbf{p}} \left| I_t(\mathbf{p}) - I_{t+1}(\mathbf{p} + \mathbf{w}_{\mathbf{p}}) \right|^2 + \lambda \sum_{\mathbf{q} \in N_{\mathbf{p}}} \left| \mathbf{w}_{\mathbf{p}} - \mathbf{w}_{\mathbf{q}} \right|^2$$

Horn & Schunck

Input

Horn & Schunck

Ground truth

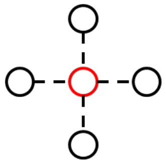


Color key
[Baker *et al.* IJCV'11]

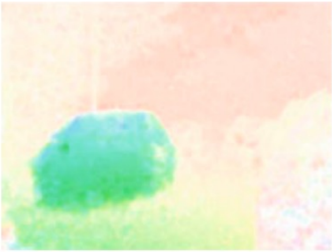
Improving Horn & Schunck

[Sun *et al.* CVPR’10, IJCV’14]

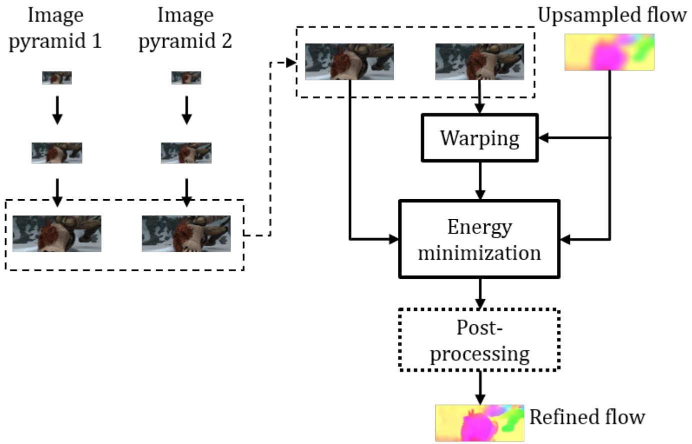
$$E(\mathbf{w}) = \sum_{\mathbf{p}} \left|I_t(\mathbf{p}) - I_{t+1}(\mathbf{p} + \mathbf{w_p})\right|^2 + \lambda \sum_{\mathbf{q} \in N_{\mathbf{p}}} \left|\mathbf{w_p} - \mathbf{w_q}\right|^2$$



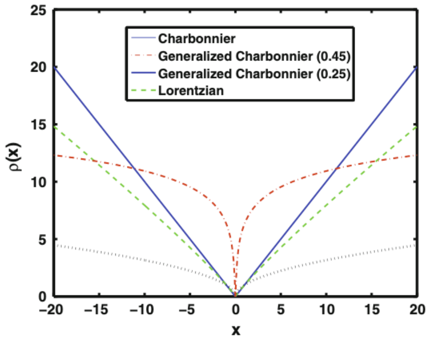
“Old”

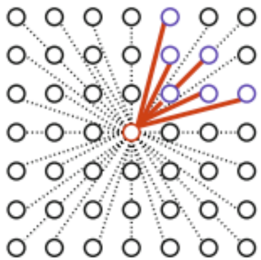


Better optimization/ implementation



Ground truth





Improving Horn & Schunck

[Sun *et al.* CVPR'10, IJCV'14]

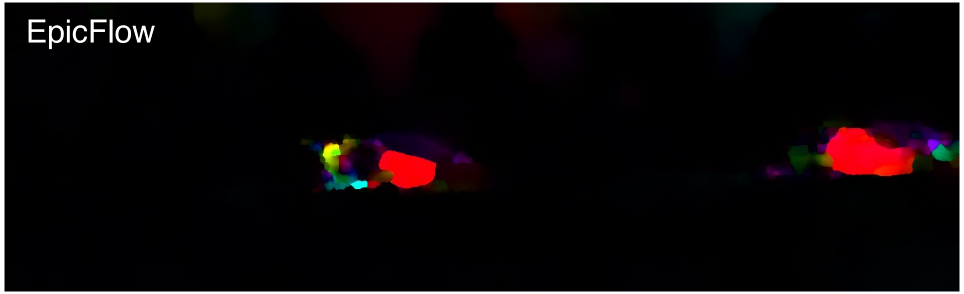
(a) "Old" **HS** [58] **(b)** "New" **HS** **(c) Classic++** **(d) Classic+NL** **(e)** Ground truth **(f)** First frame

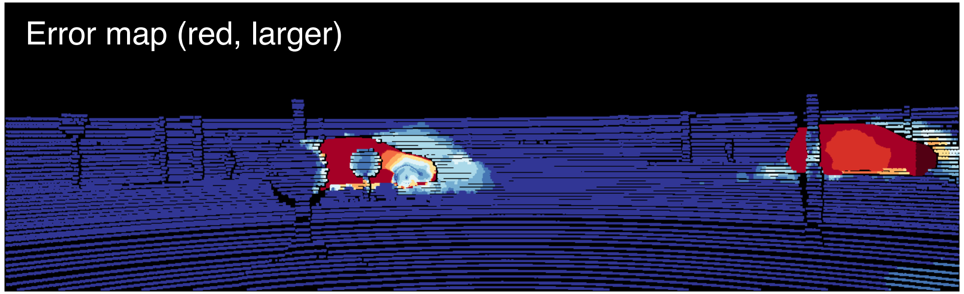
Implementation   Robust function   Non-local smoothness term

Challenges for classical methods

Large motion
Motion blur
Occlusions
Lighting changes
Noise…

Hard to modify objective function and even harder to optimize it





Content

- Classical approach
  - Constancy assumption -> matching by comparison (cost volume)

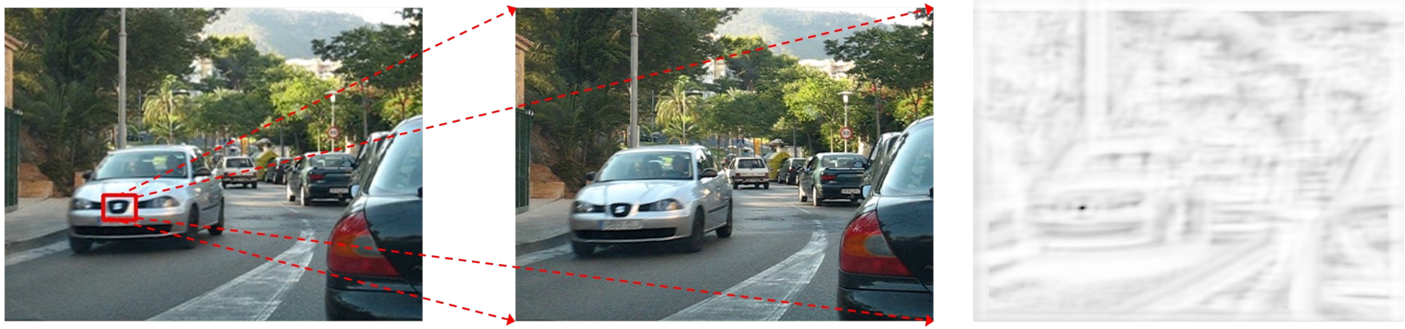
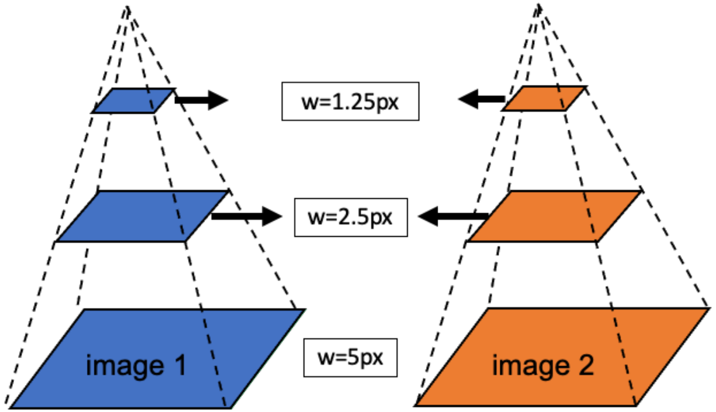


Image 1 Image 2 Cost volume

Content

- Classical approach
  - Constancy assumption -> matching by comparison (cost volume)
  - Coarse-to-fine, warping-based iterative estimation



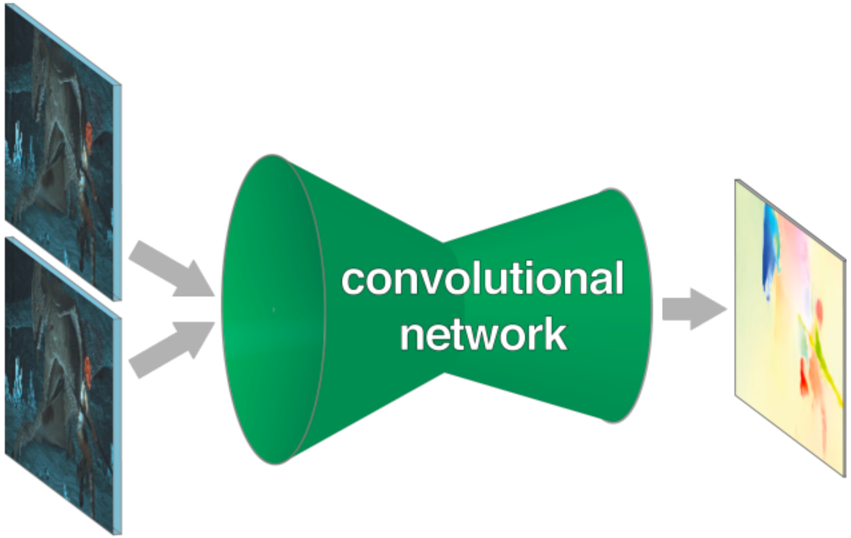
Input images *t* and *t*+1

Image *t* and warped image

Deep learning-based approach

Supervised optical flow

[Dosovitskiy *et al.* ICCV'15]

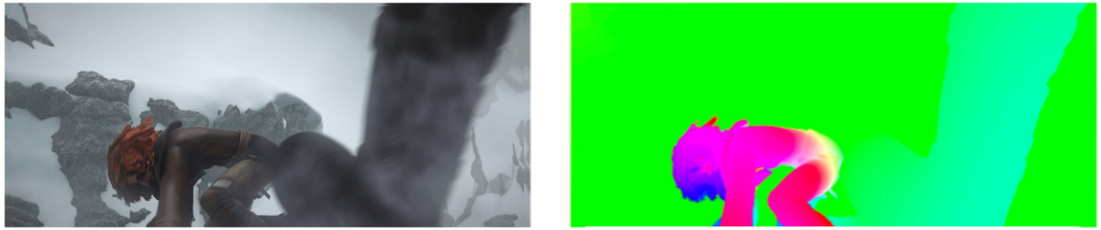


What is the training/test data?

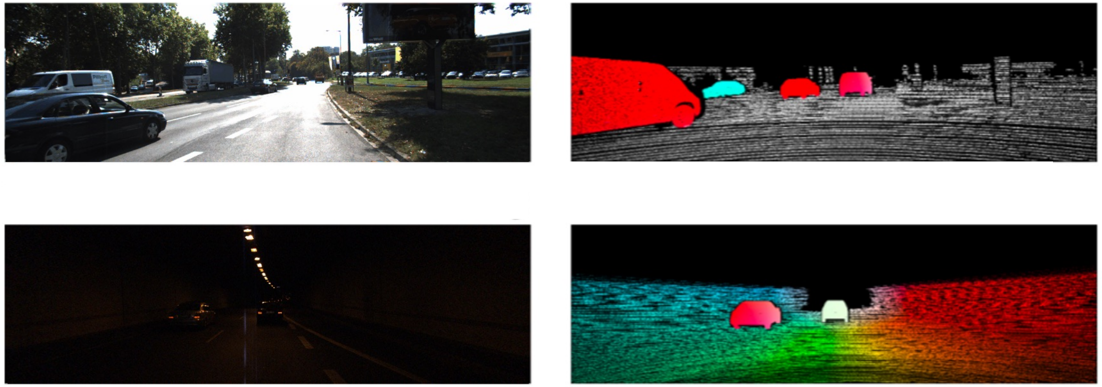

Training Data:
FlyingChairs

Two widely-used benchmarks for optical flow

Sintel (Blender movie)

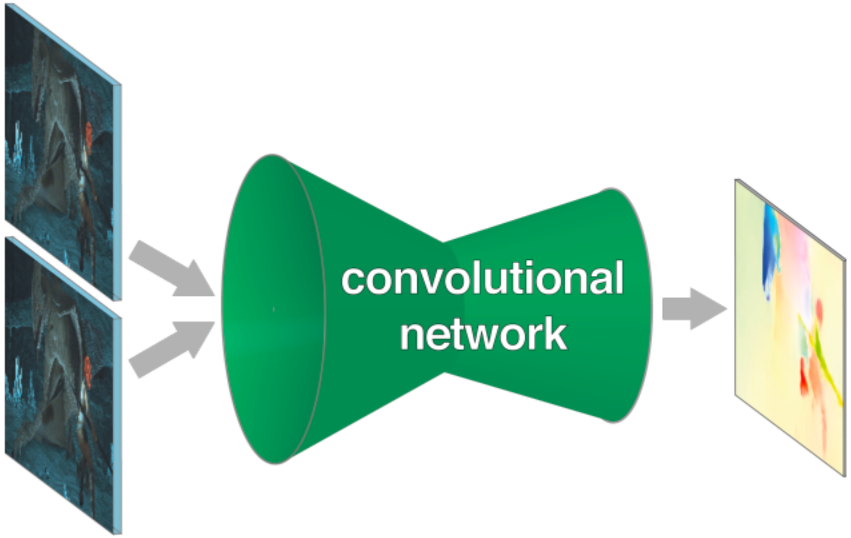


KITTI (driving)



Supervised optical flow

[Dosovitskiy *et al.* ICCV'15]

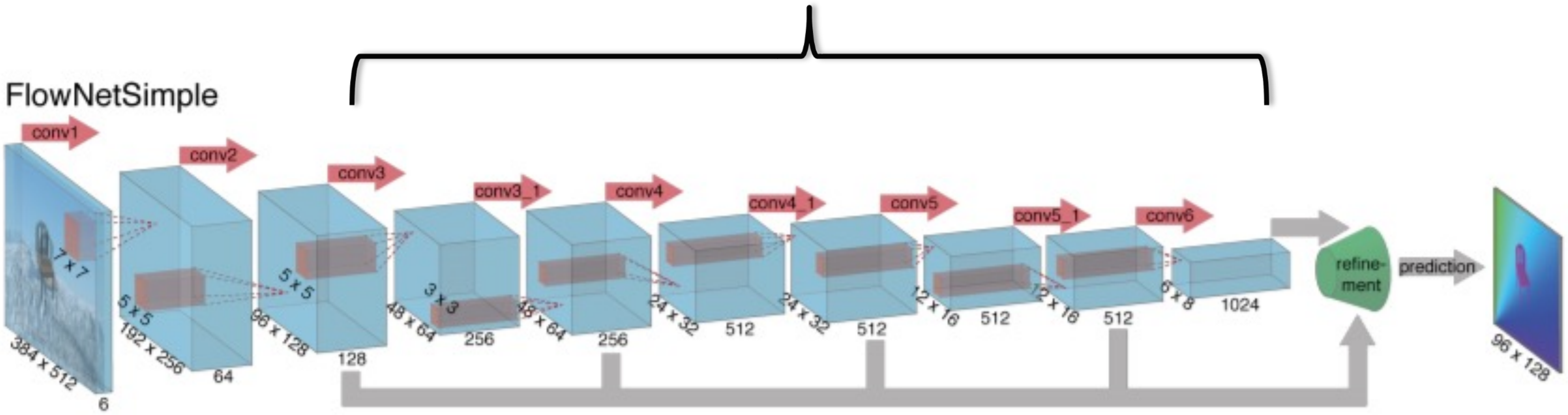


What is the network/architecture?

FlowNetS(imple): Mapping from images to flow

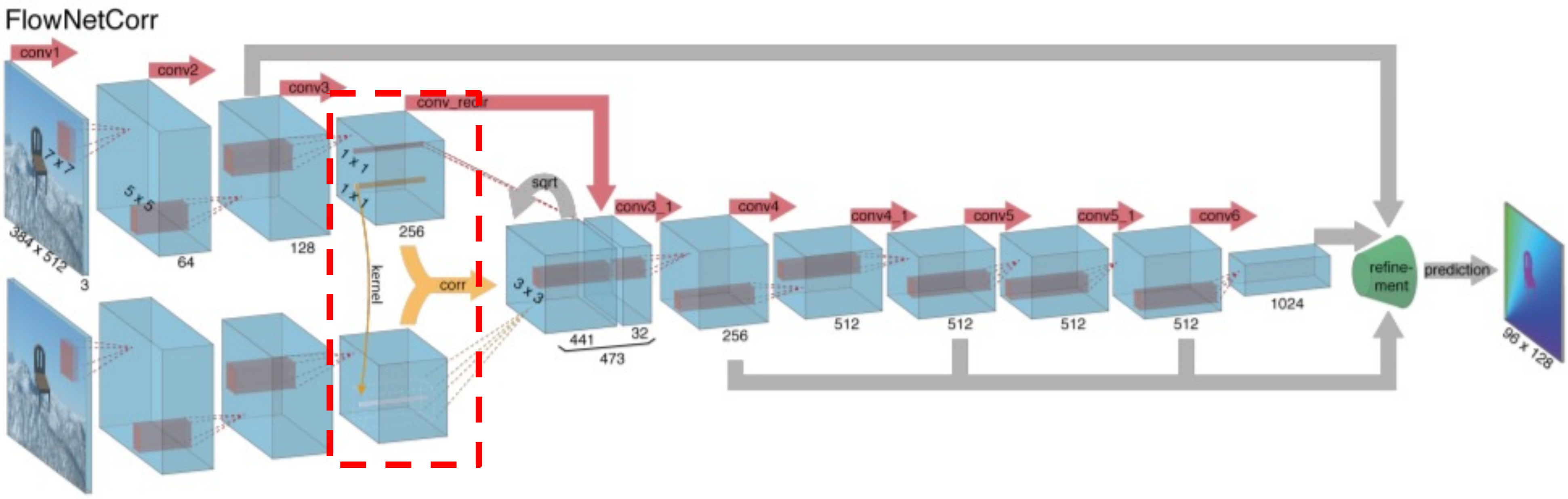
[Dosovitskiy *et al.* ICCV'15]

U-Net architecture



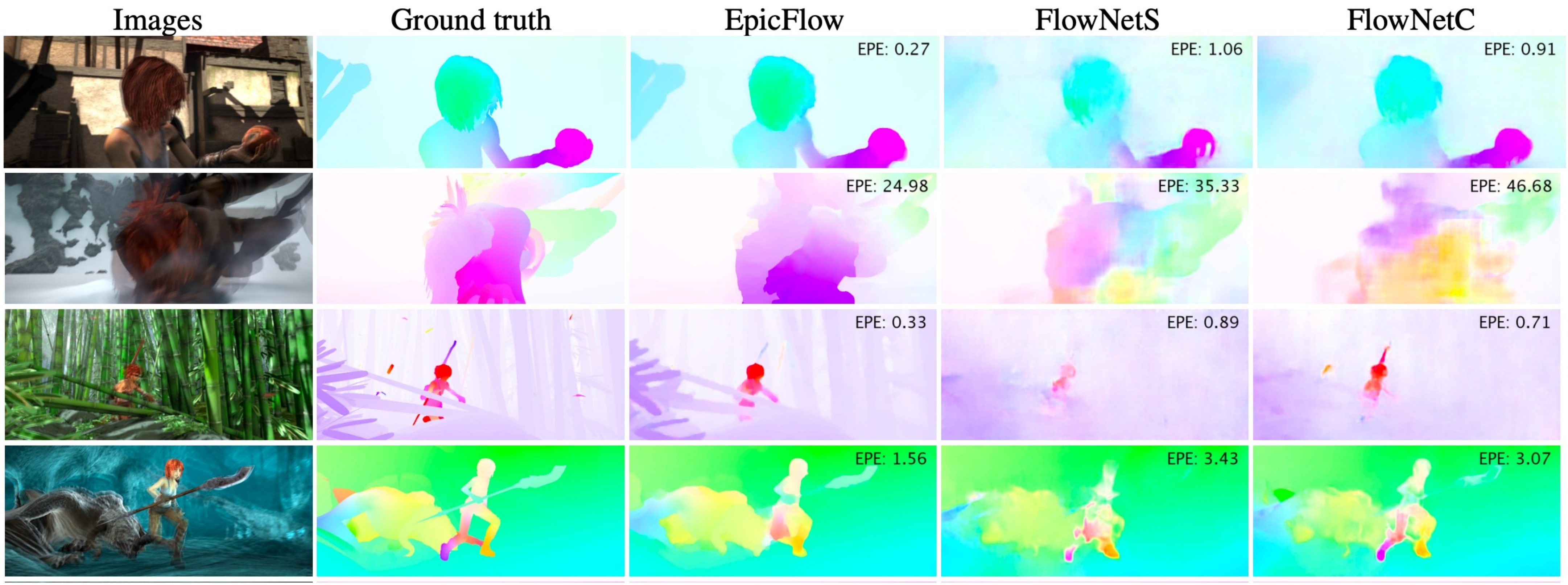
FlowNetC(orrelation): Compare features

[Dosovitskiy *et al.* ICCV'15]



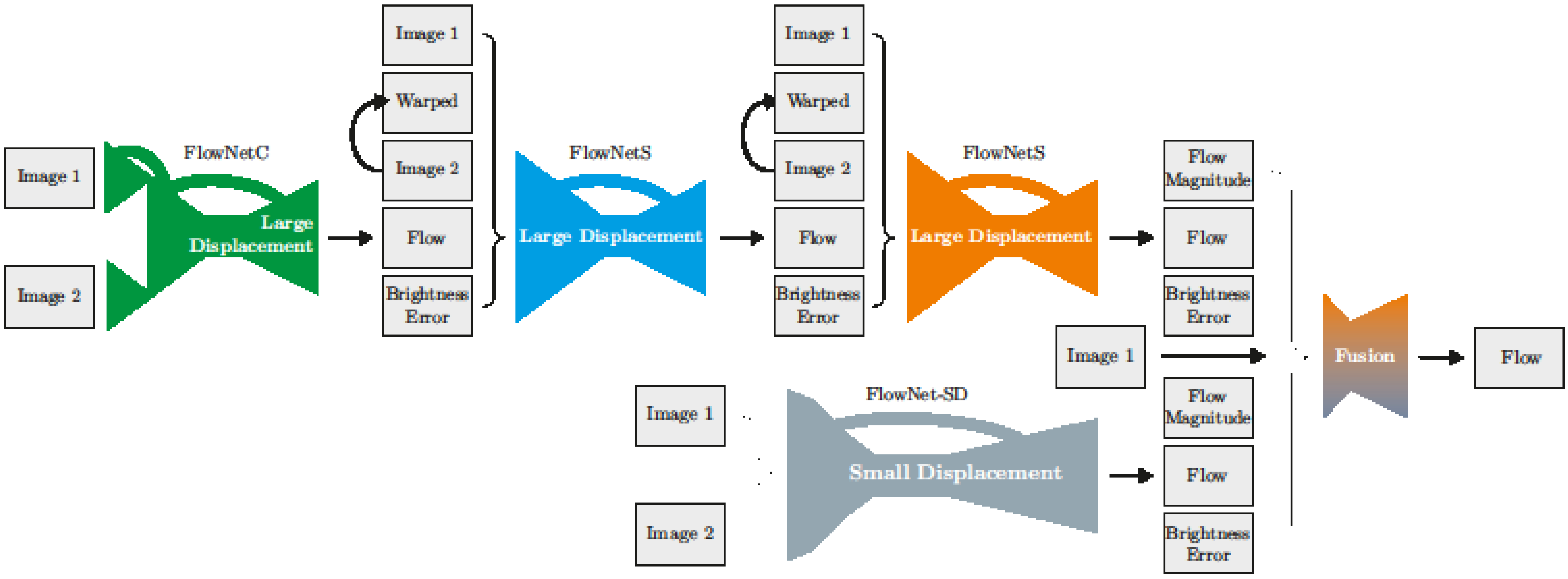
Promising but behind contemporary state of the art

[Dosovitskiy *et al.* ICCV'15]

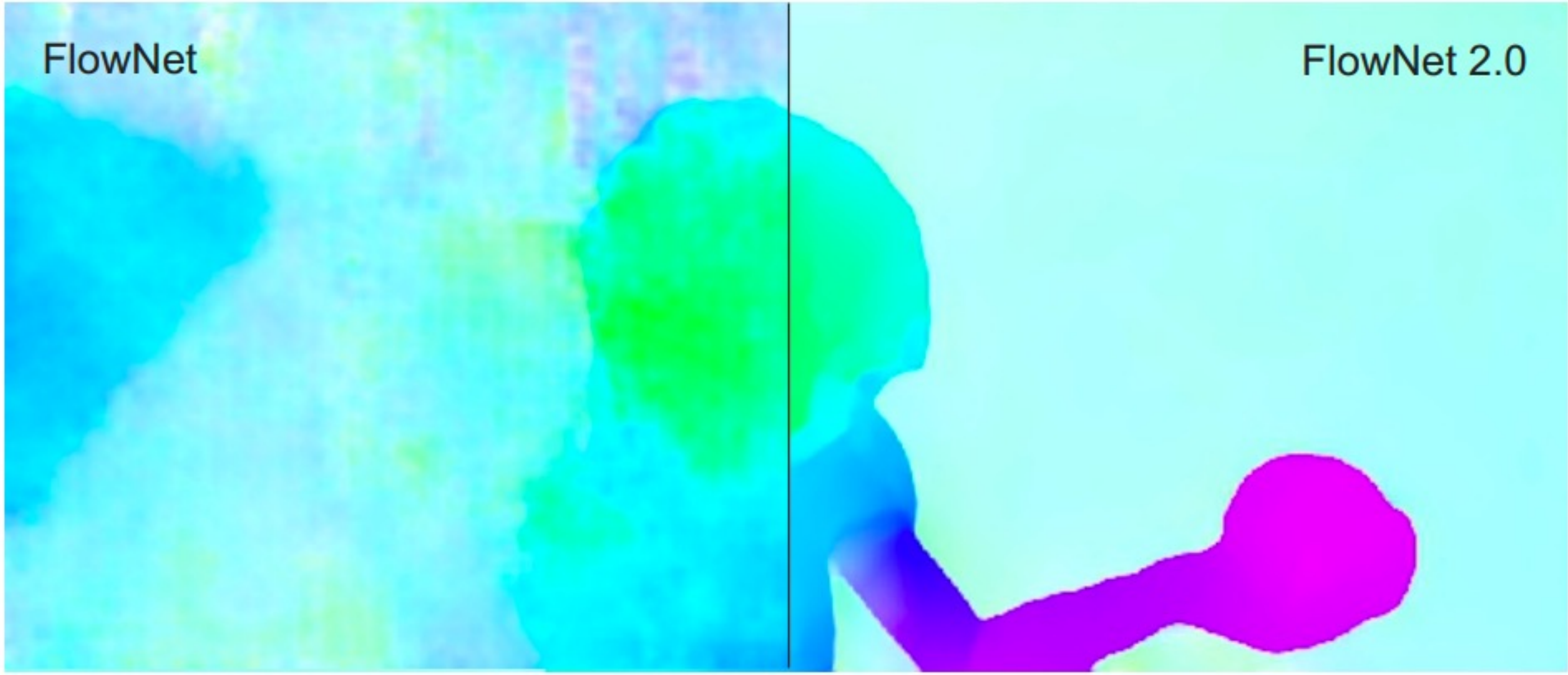


FlowNet2: Scaling up by stacking up FlowNetS/C

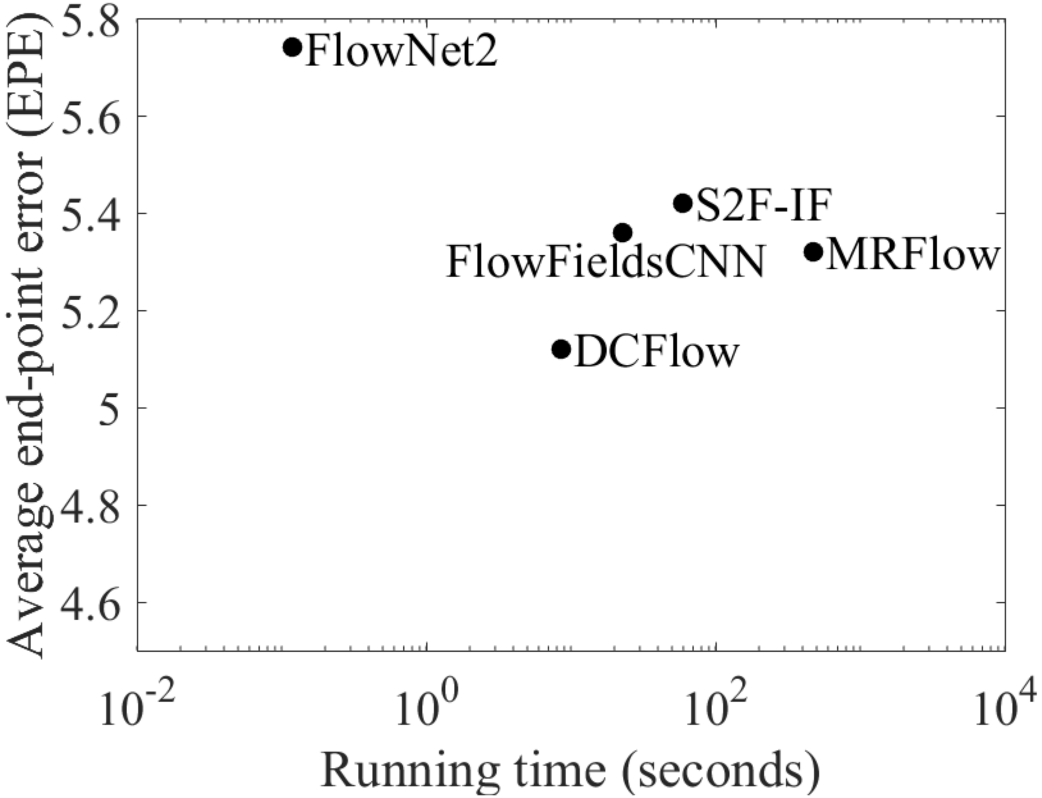
[Ilg *et al*. CVPR'17]



Significant improvement

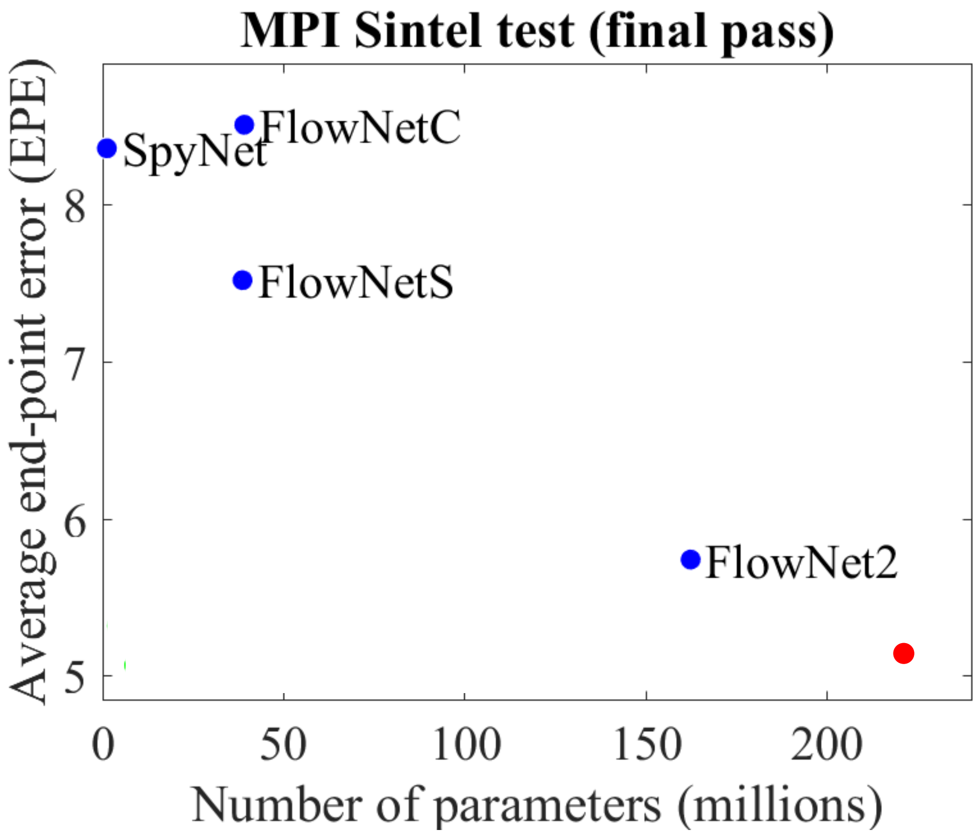


Trade-off between accuracy and running time



FlowNet2: Ilg *et al*. CVPR’17
S2F-IF: Yang & Soatto CVPR’17
FlowFieldsCNN: Bailer *et al*. CVPR’17
MRFlow: Wulff *et al*. CVPR’17
DCFlow: Xu *et al*. CVPR’17

Trade-off between accuracy and size for CNN methods



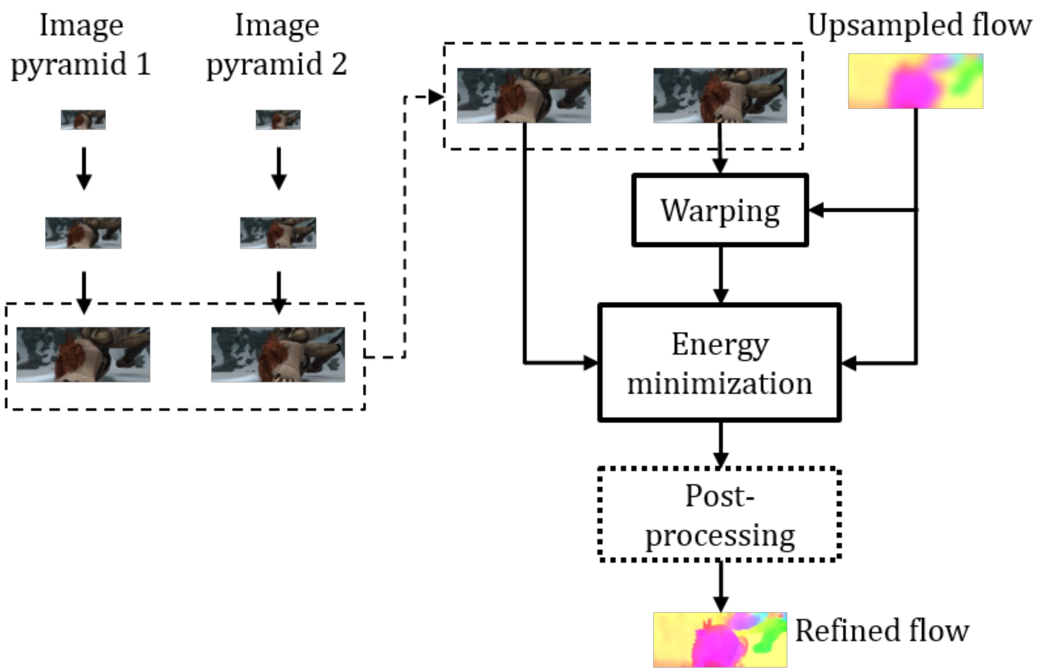
FlowNetS/C: Dosovitskiy *et al.* ICCV'15
FlowNet2: Ilg *et al*. CVPR'17
SpyNet: Ranjan & Black CVPR'17

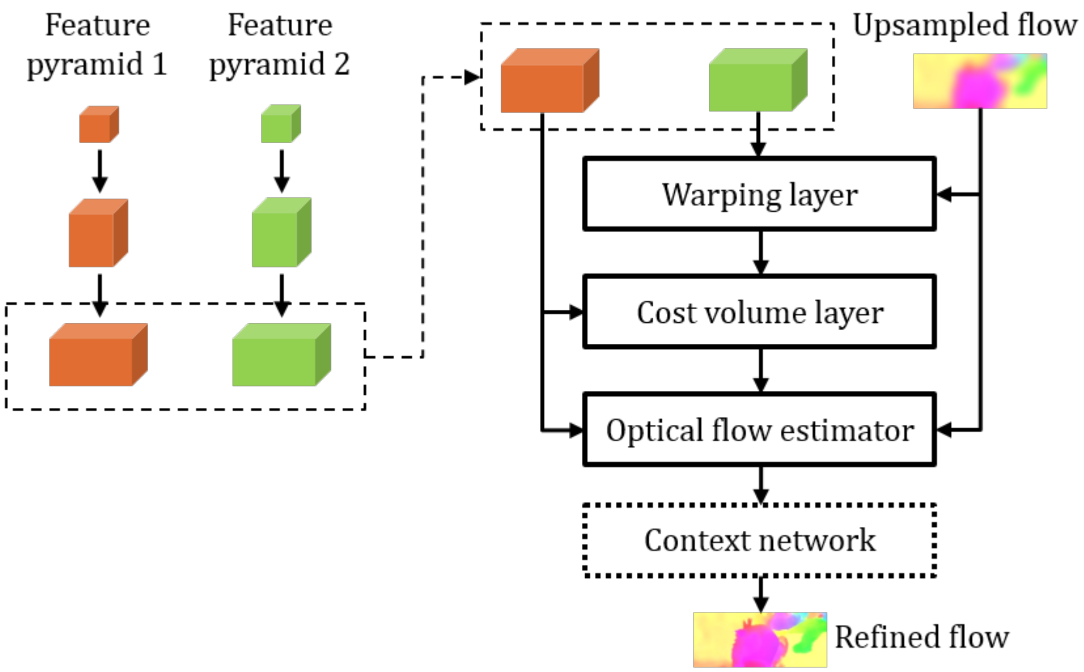
PWC-Net: Sun et al. CVPR'18

Pyramid, warping, & cost volume, not PricewaterhouseCoopers☺

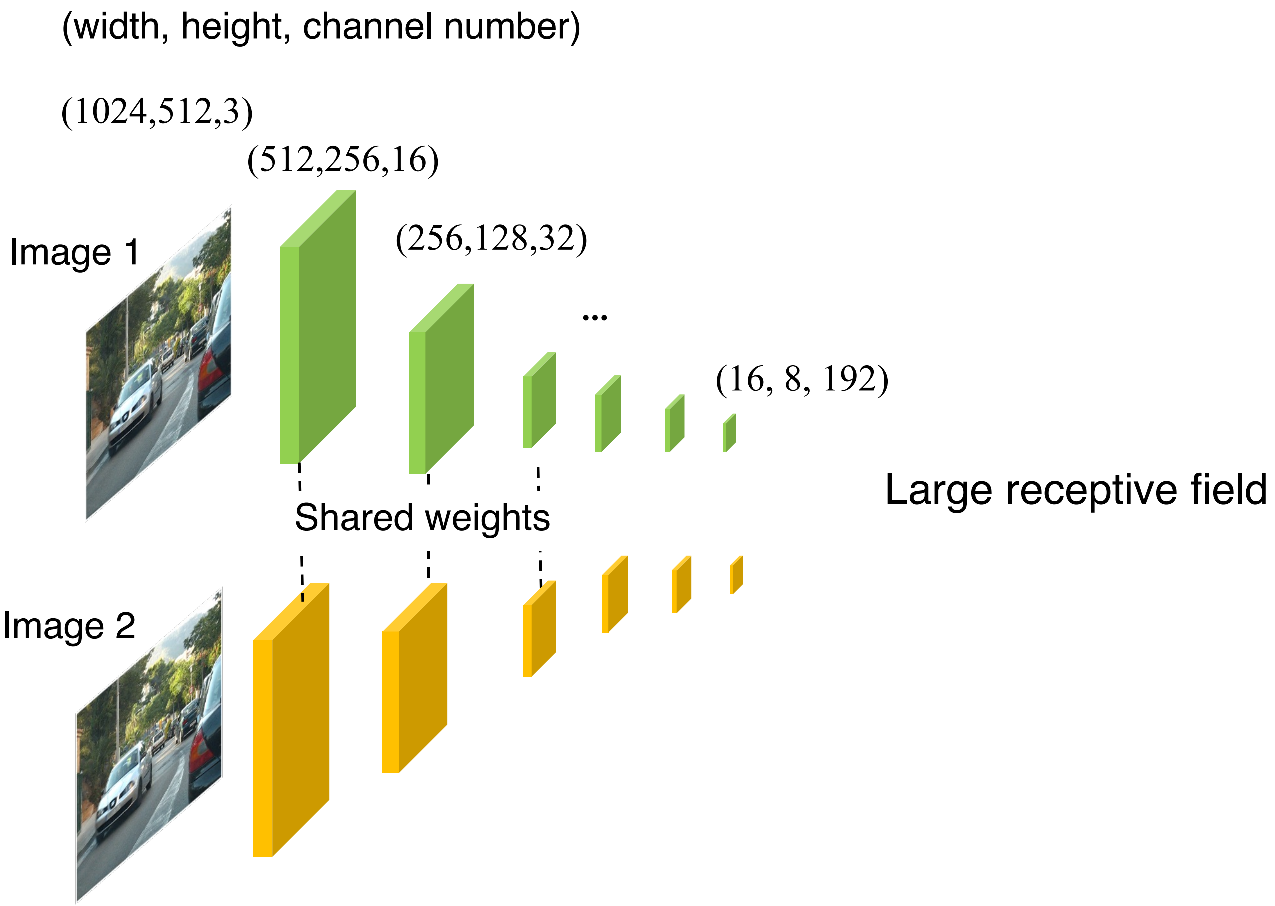
Even bigger model?

Inspired by classical approach





Pyramid of learnable features

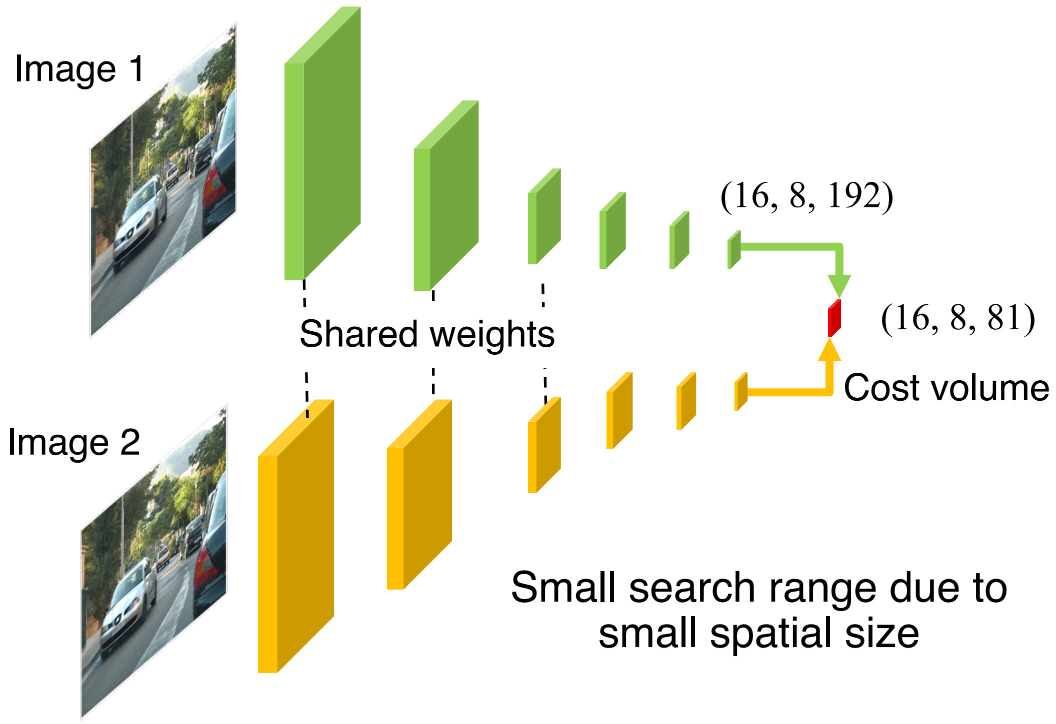


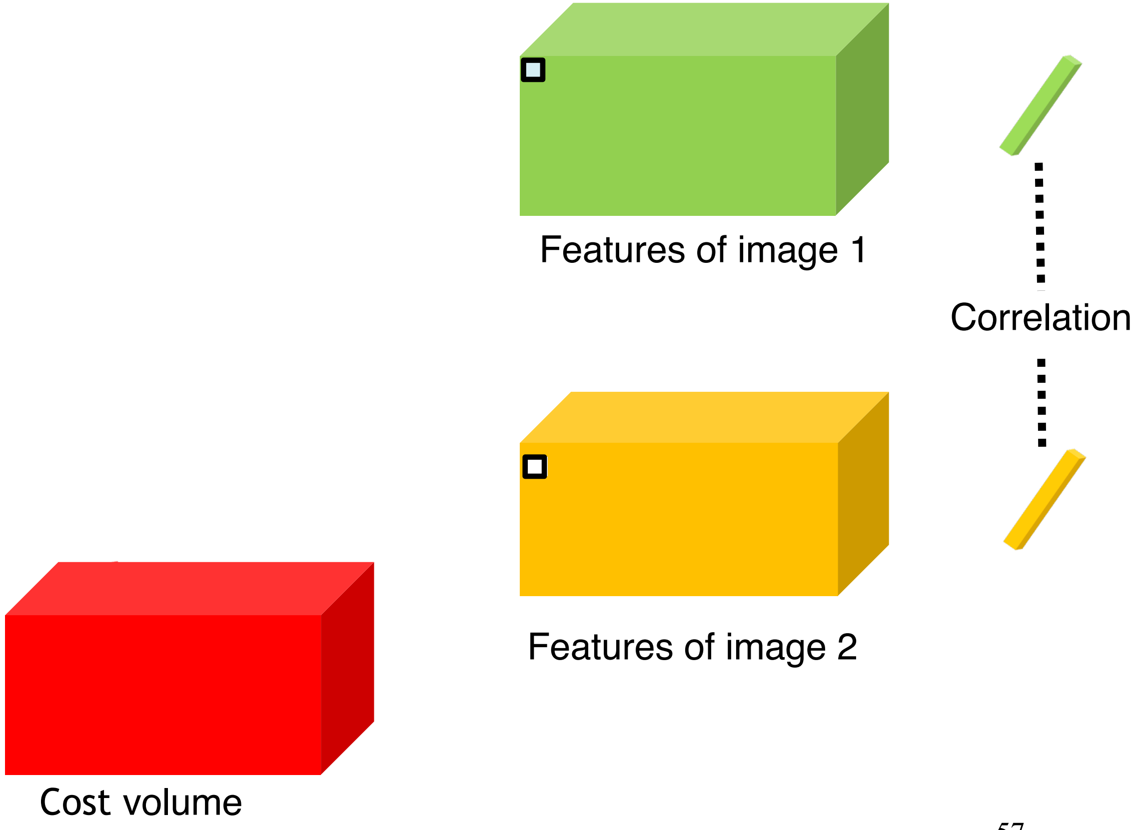
Compute cost volume by correlation

[Dosovitskiy *et al.* FlowNet ICCV'15]

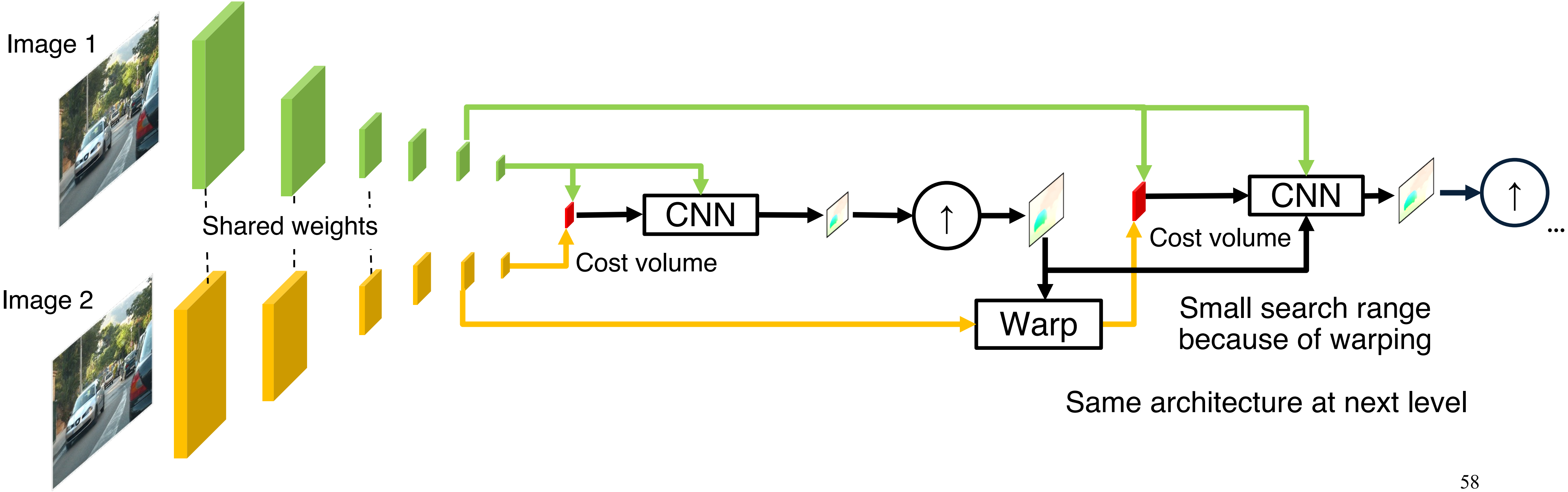
(width, height, channel number)

(1024,512,3)

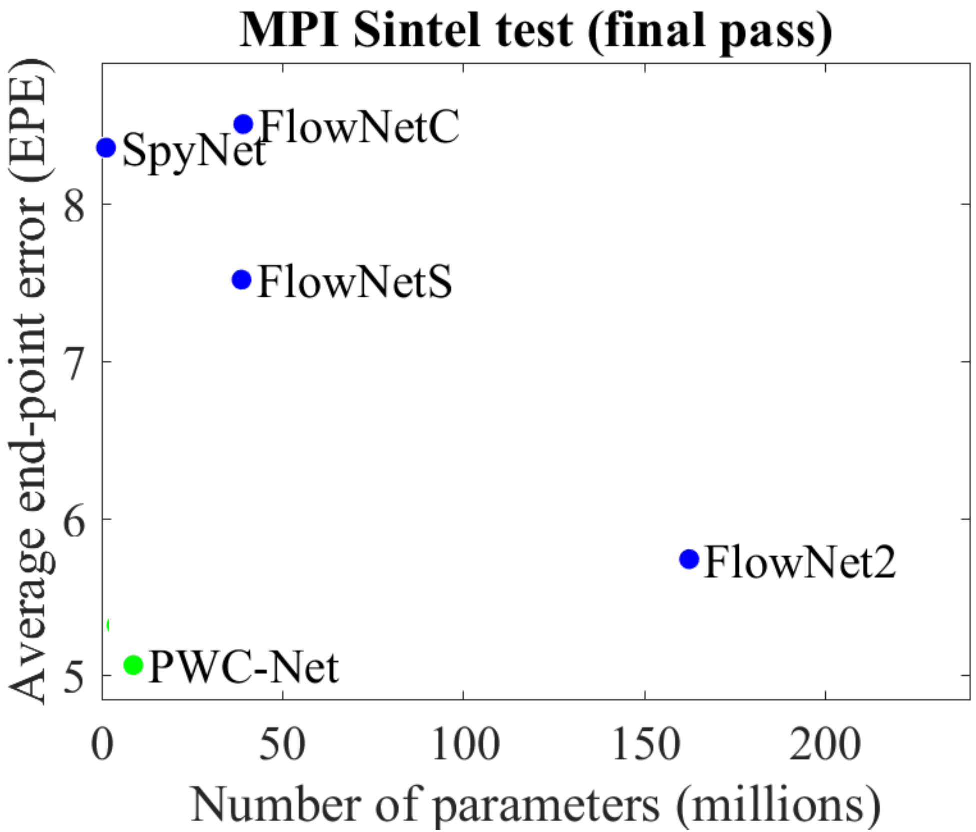




Mapping cost volume to optical flow



Architectures matter



Flow Leaderboard

Final results of the ROB 2018 Challenge. New submissions will be accepted after CVPR 2018.

	Method	Middlebury (Detailed subrankings)	KITTI (Detailed subrankings)	MPI Sintel (Detailed subrankings)	HD1K (Detailed subrankings)
1	PWC-Net_ROB	2	2	2	1
	PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume [Project page] - Submitted by Deqing Sun (NVIDIA)				
2	ProFlow_ROB	1	5	1	3
	Submitted by Daniel Maurer (University of Stuttgart)				
3	LFNet_ROB	6	1	5	4
	Submitted by Anonymous				
4	AugFNG_ROB	8	3	3	2
	Submitted by Anonymous				
4	FF++_ROB	3	4	4	5
	FlowFields++: Accurate Optical Flow Correspondences Meet Robust Interpolation [Project page] - Submitted by René Schuster (DFKI)				
6	DMF_ROB	4	7	6	7
	DeepFlow: Large displacement optical flow with deep matching [Project page] - Submitted by Alexander Brock (HCI)				
6	ResPWCR_ROB	5	6	7	6
	Submitted by Anonymous				
8	WOLF_ROB	7	8	8	8
	Submitted by Anonymous				

Visual results on KITTI video sequence

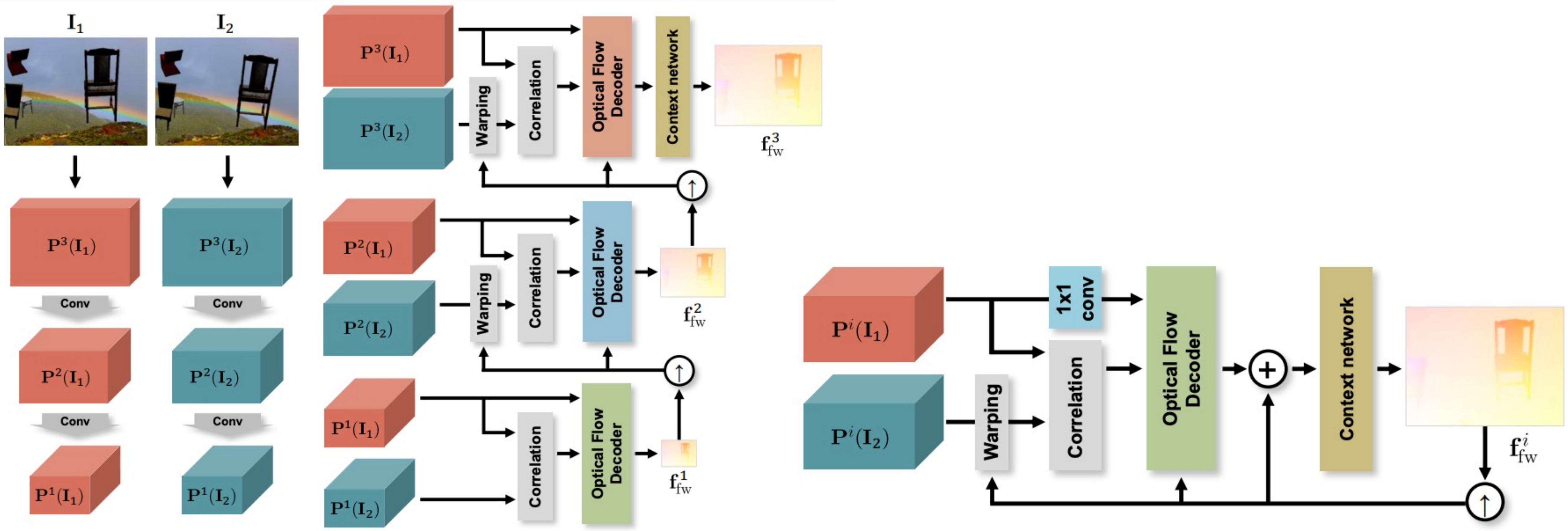
Tensorflow code

Caffe & PyTorch code

35 fps for Sintel (1024x448) resolution on NVIDIA Pascal TitanX

Improvement: Iterative Residual Refinement (IRR)

[Hur and Roth CVPR 2019]



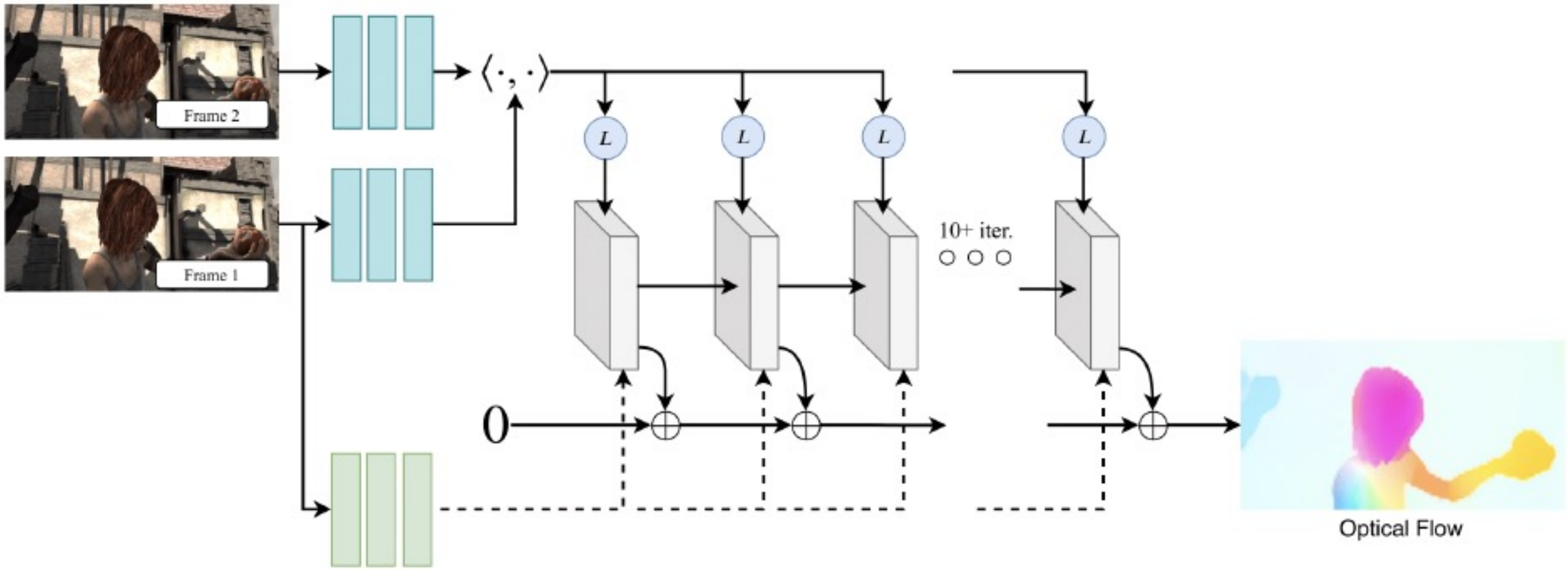
(a) Original PWC-Net [52].

(b) Our IRR version of PWC-Net.

Figure credit: Hur and Roth

RAFT: Recurrent All-pairs Field Transforms

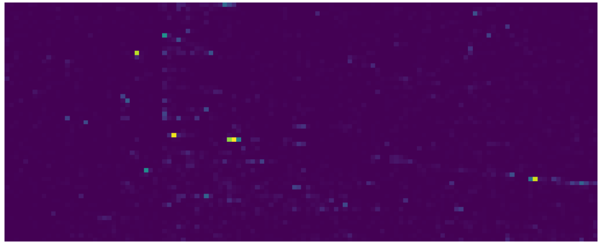
[Teed and Deng ECCV 2020 Best paper]

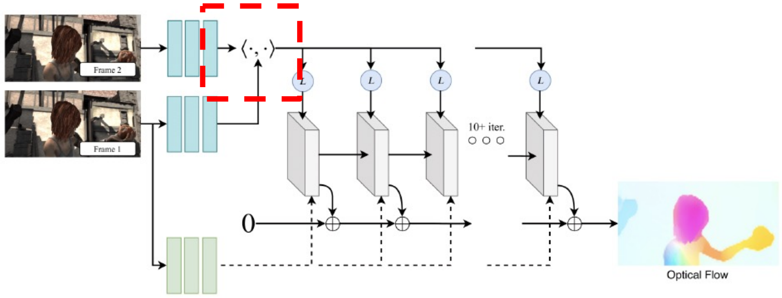


Slide credit: Teed and Deng

All-pairs visual similarity (cost volume)

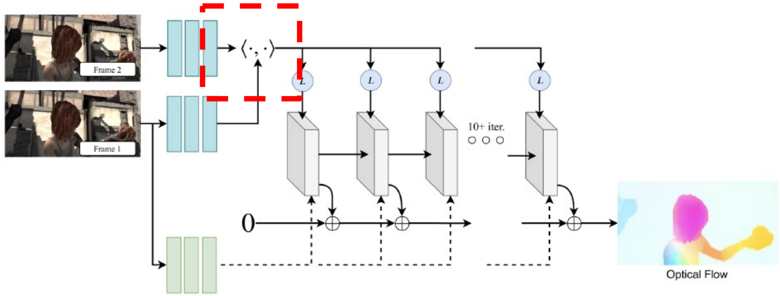
Inner product/correlation between features

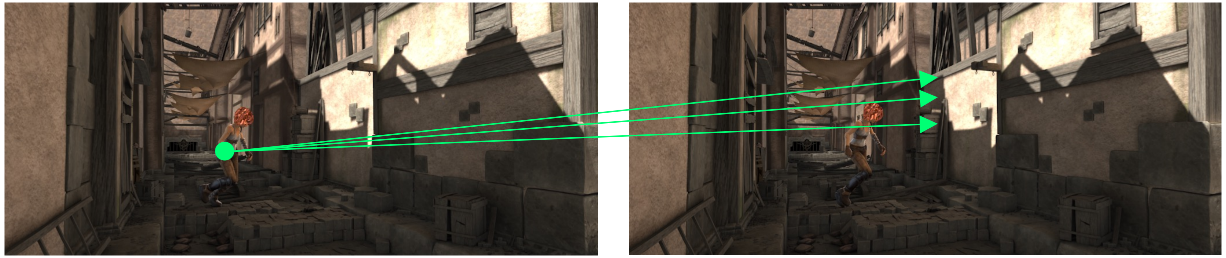


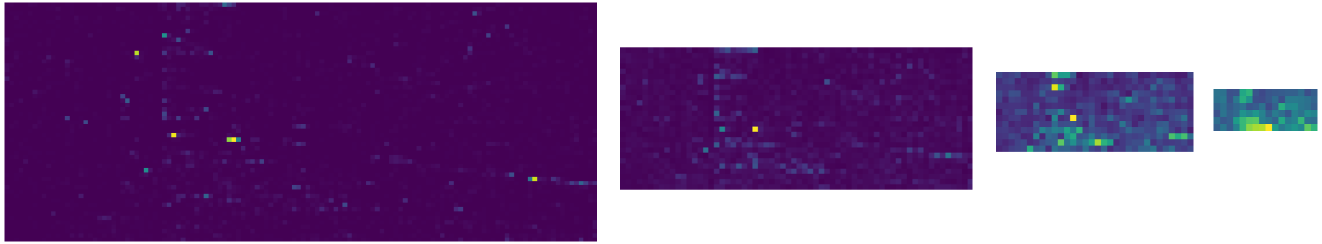


Cost volume pyramid

Spatial pooling

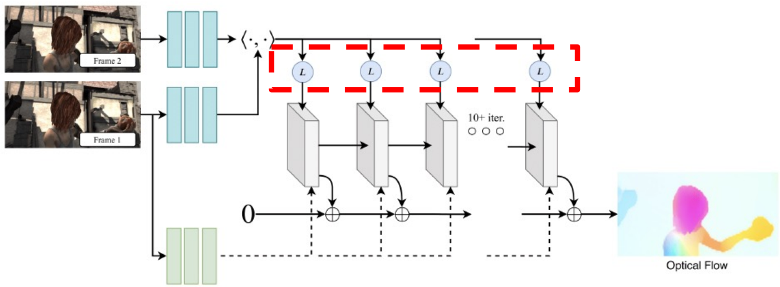


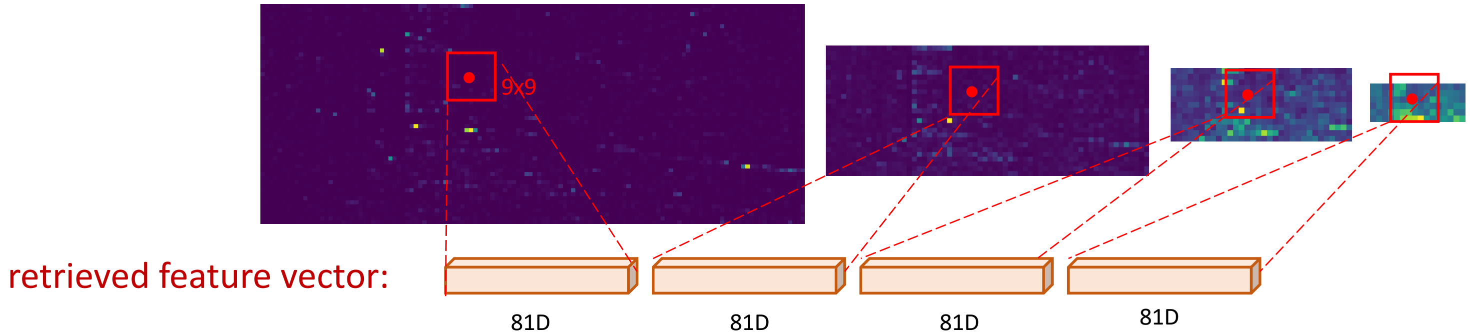




Look up cost volume

Retrieve using current motion estimates

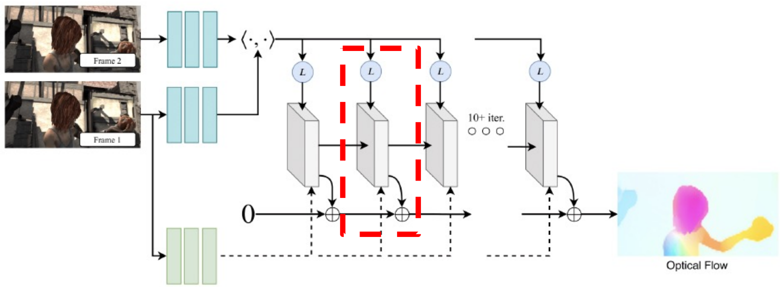


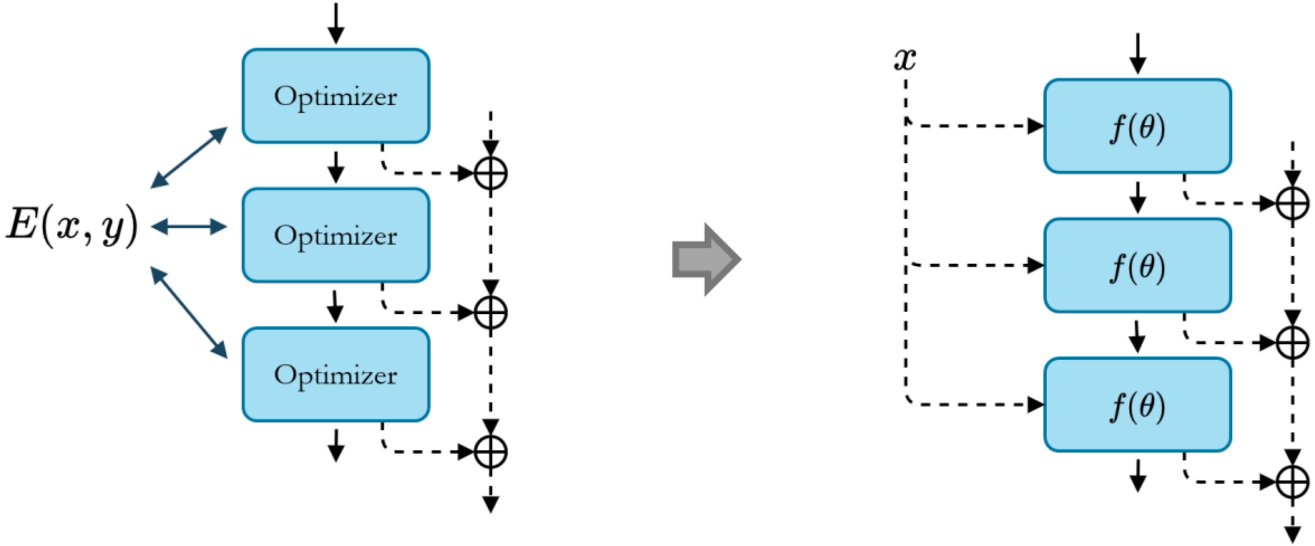


cues on how good the current flow is and where are better similarities

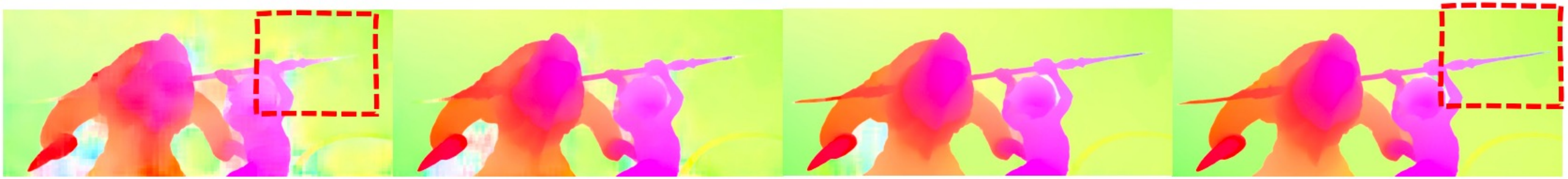
Recurrent update

Like classical optimization algorithms





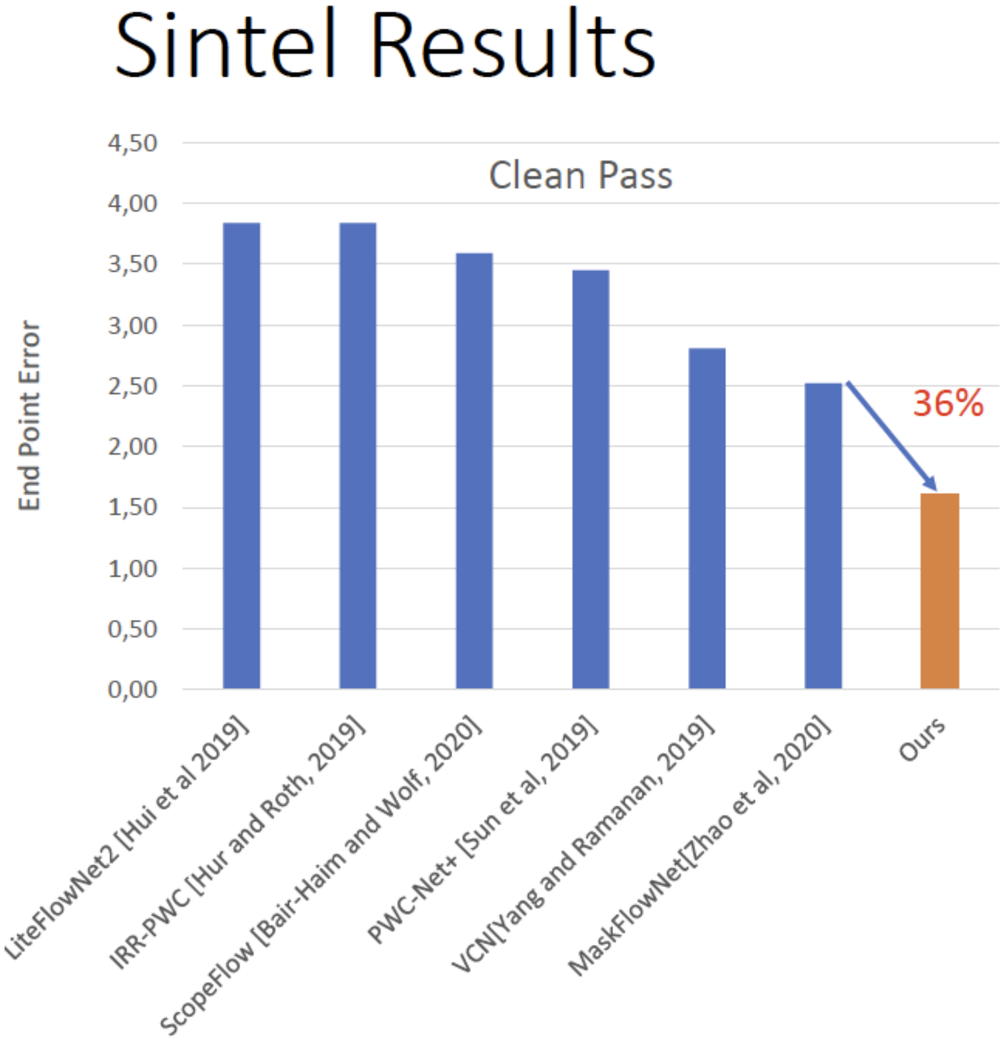
Recurrent update



1 Iteration   2 Iterations   5 Iterations   32 Iterations

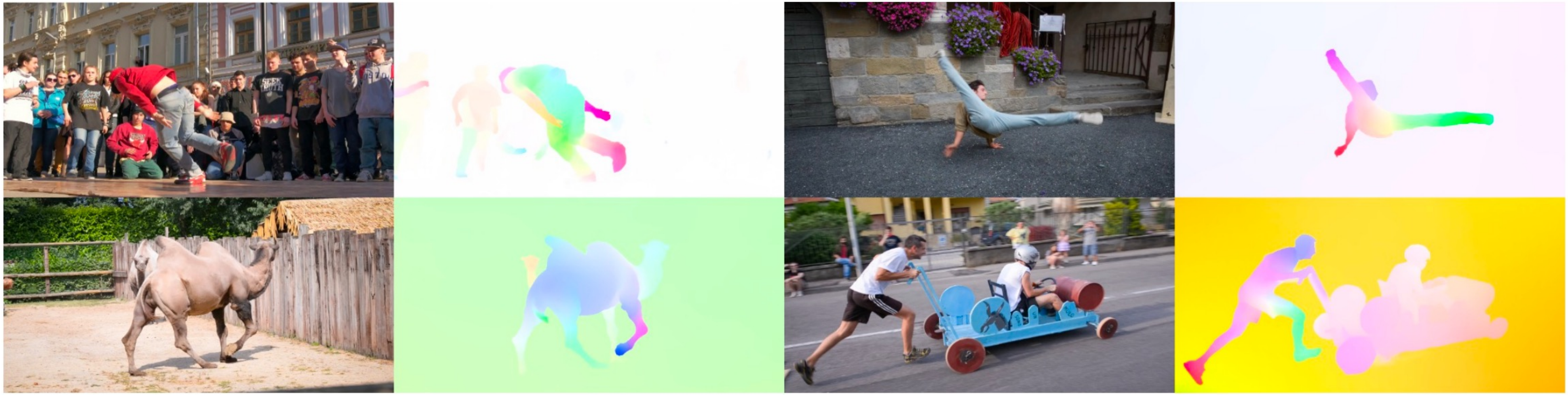
Significant improvement over prior art

[Teed and Deng ECCV 2020 Best paper]



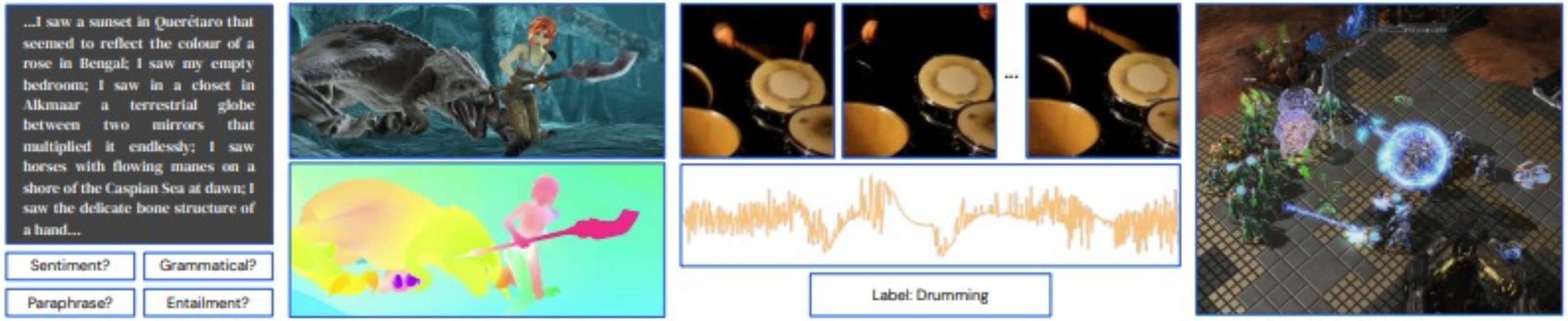
Slide credit: Teed and Deng

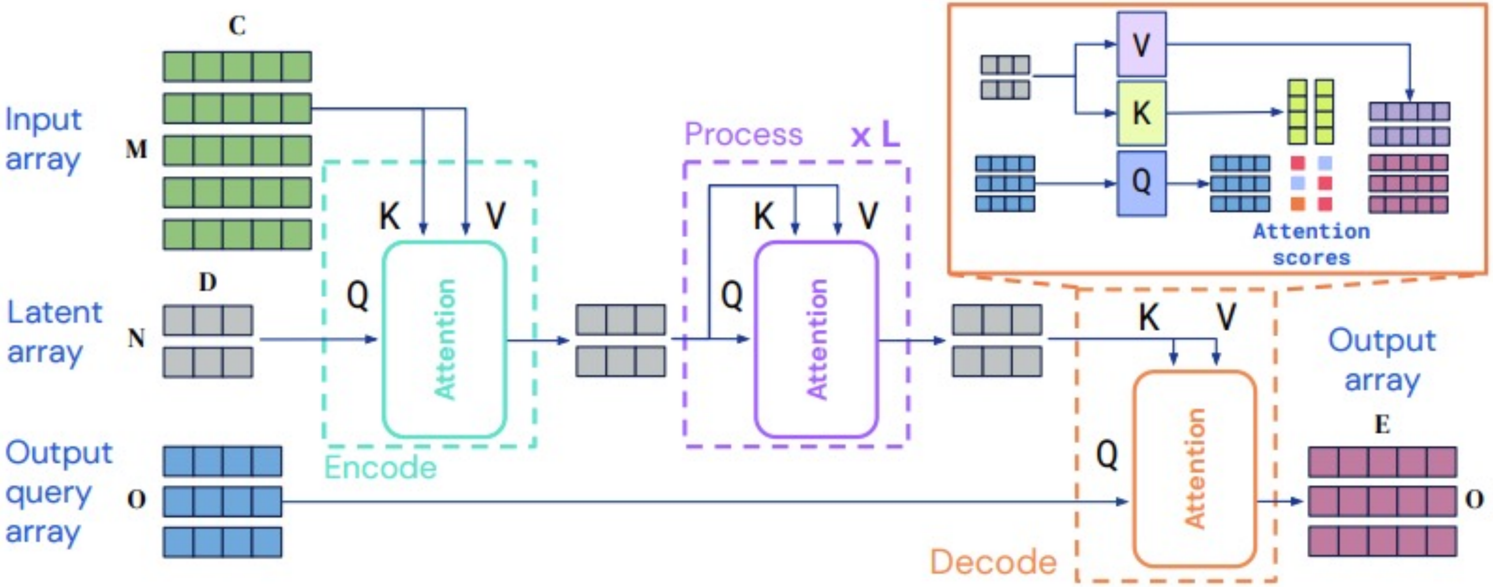
Visual results on Davis (real-world)



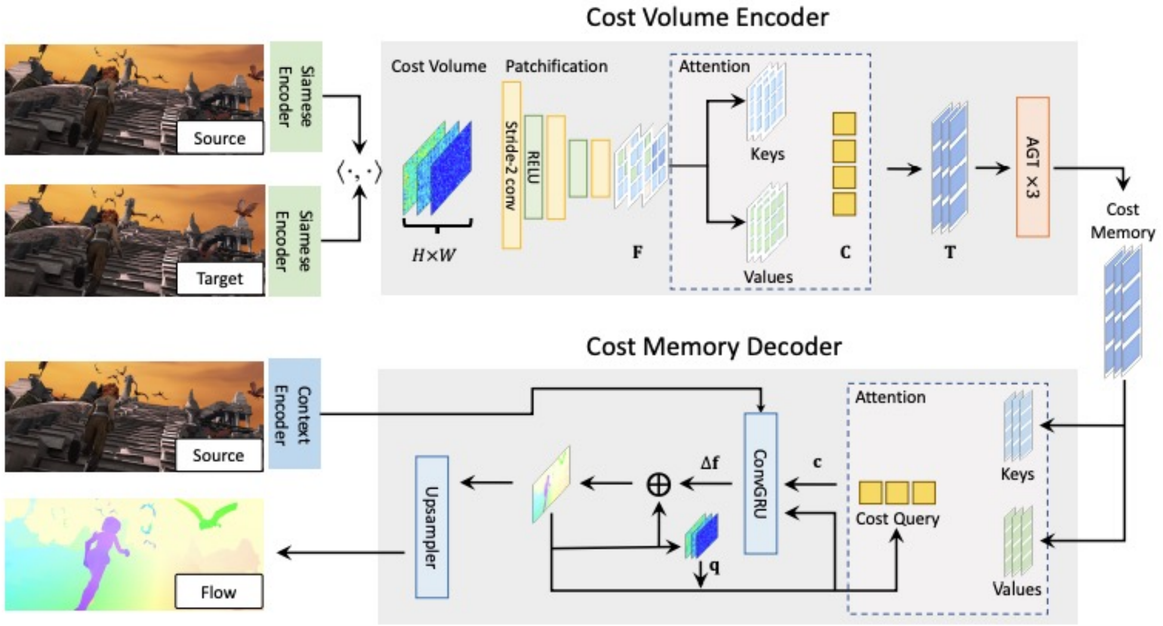
Recent development: Attention/transformer

[Perceiver IO: A General Architecture for Structured Inputs & Outputs](). ICLR ‘22

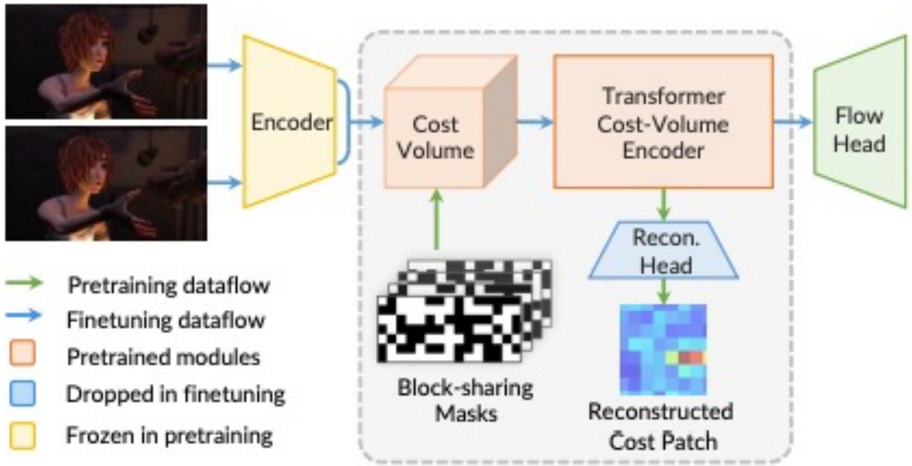




Flowformer: A transformer architecture for optical flow



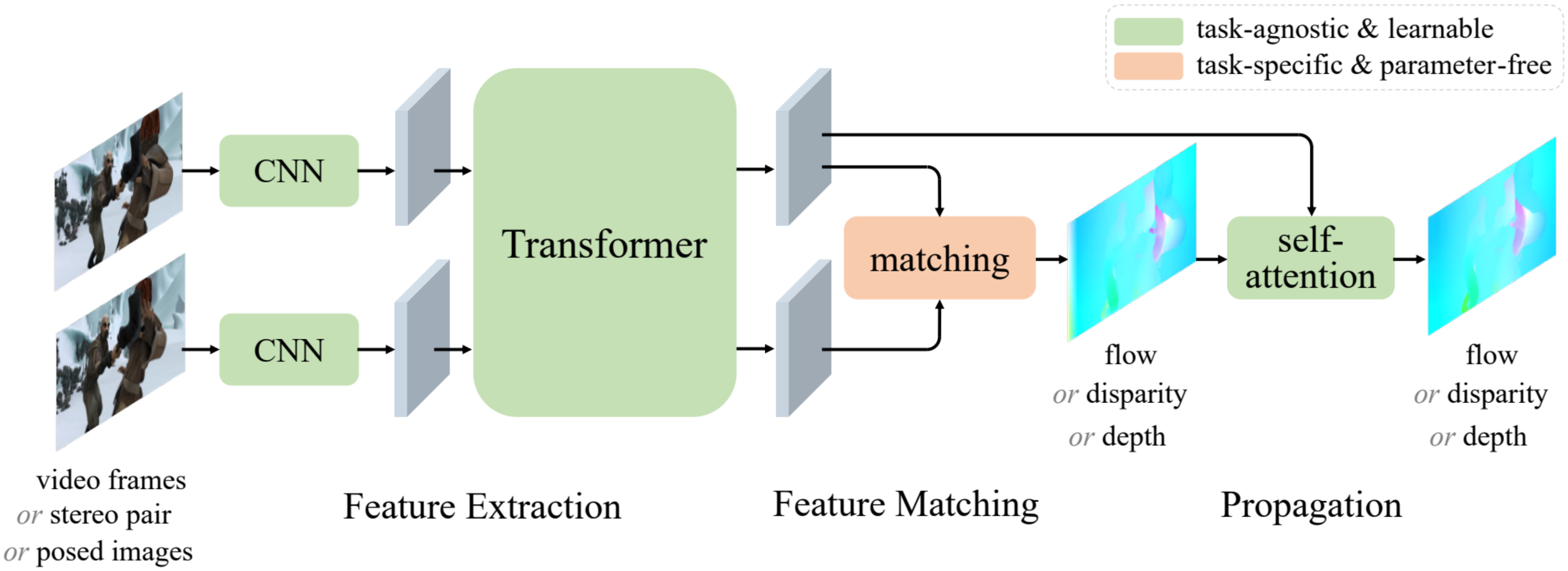
[Flowformer] [Huang *et al.* ECCV '22]



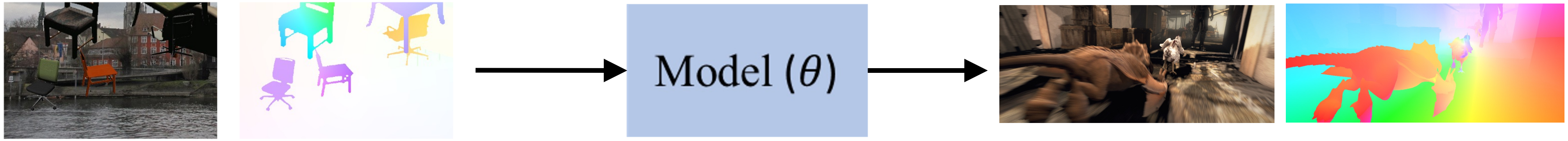
[Flowformer++] [Shi *et al.* arXiv '23]

GM-Flow: Unifying flow, stereo and depth estimation

[Xu *et al.* arXiv '23]



Is architecture all we need?



FlyingChairs

Train

Evaluate

Learning Data

“FlyingChairs” manually designed in 2015

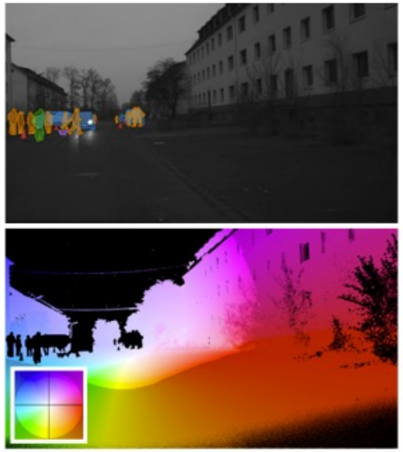
[Dosovitskiy *et al.* ICCV’15]

“FlyingChairs” more effective than later datasets

“FlyingThings3D” [Mayer et al. 2016]

Virtual KITTI [Gaidon et al. 2016]

Playing for benchmark [Richter et al. 2017]



HD1K [Kondermann et al. 2016]

Multi-human [Ranjan et al. 2020]

Refresh [Lv et al. 2018]

…

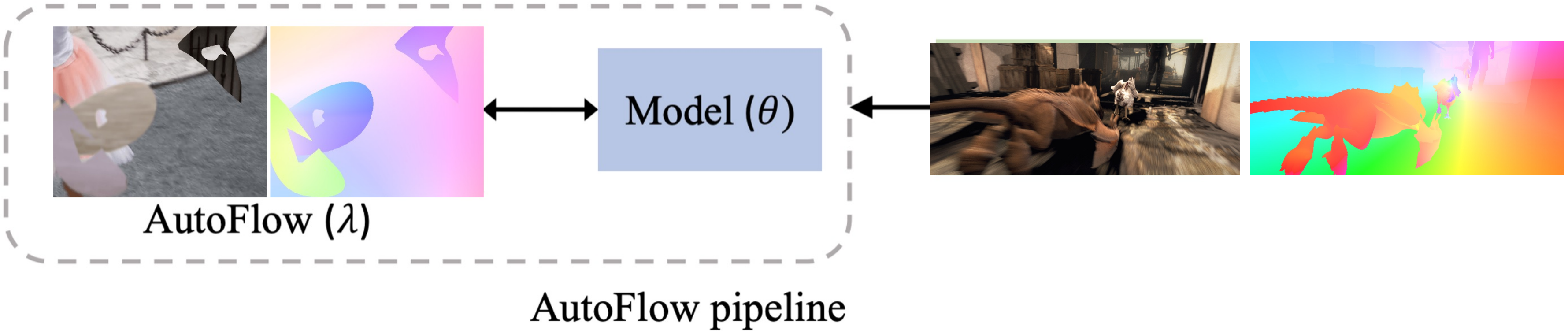
Many important and interesting questions unanswered

- How realistic should the rendering be?
- Why does FlyingChairs work so well?
- Should we carefully match the motion statistics of Sintel?
- Are thin structures/fine motion details of FlyingChairs critical?

…

What is the objective for rendering training data?

Optimize the performance of a network on a target dataset



Jointly render data and train model

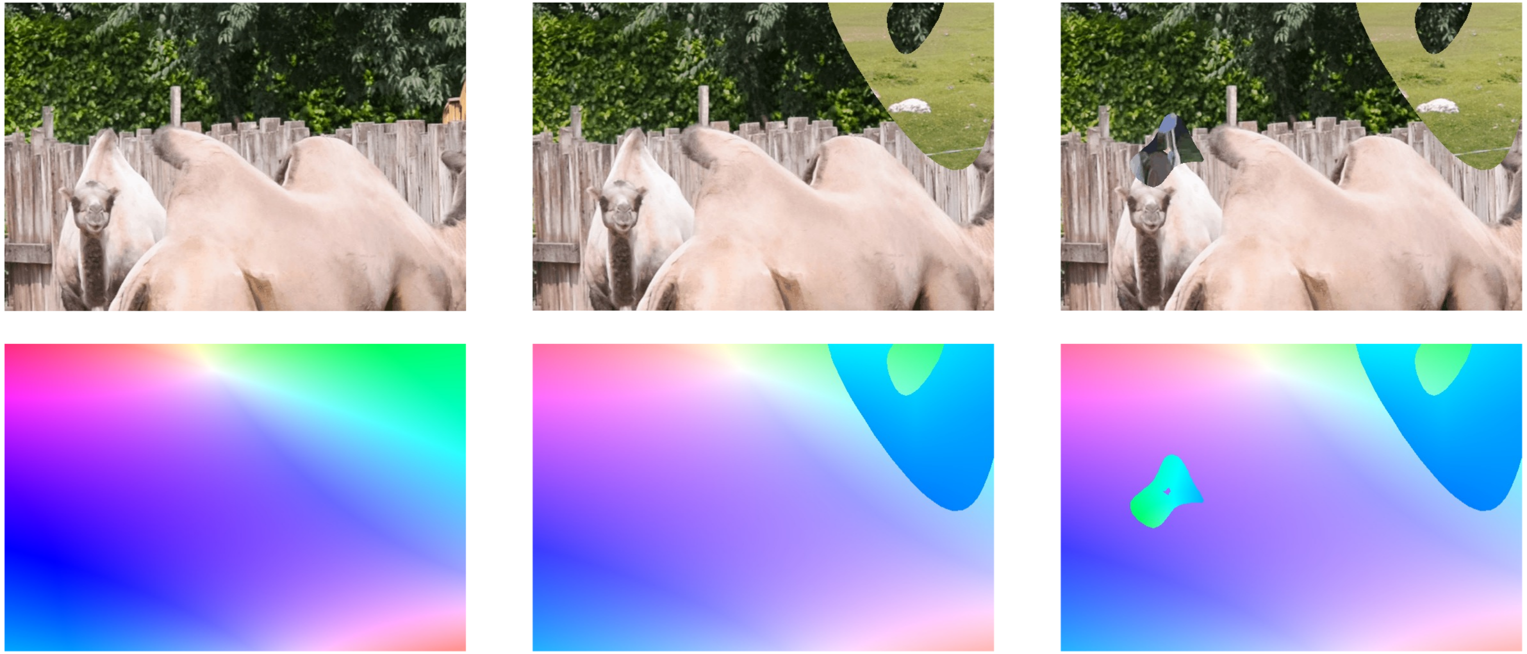
Sun *et al.* AutoFlow: Learning a Better Training Set for Optical Flow, CVPR'21 Oral.

How simple can the rendering be?

Start from the simplest rendering pipeline: 2D layered model

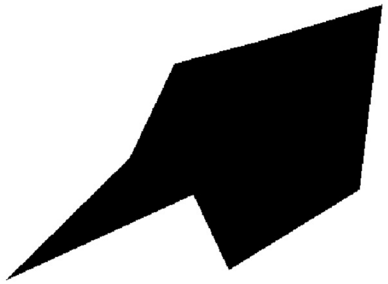
Images

Motion



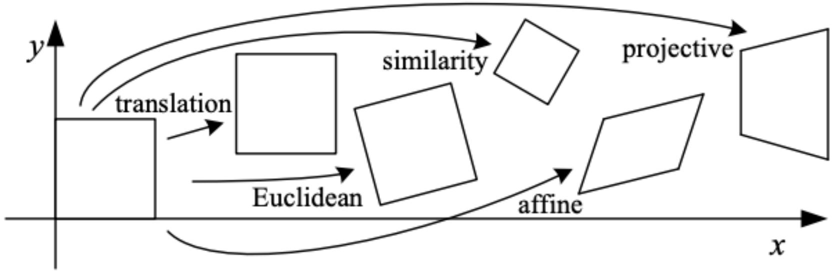
Background   + 1 foreground objects   + 2 foreground objects   . . .

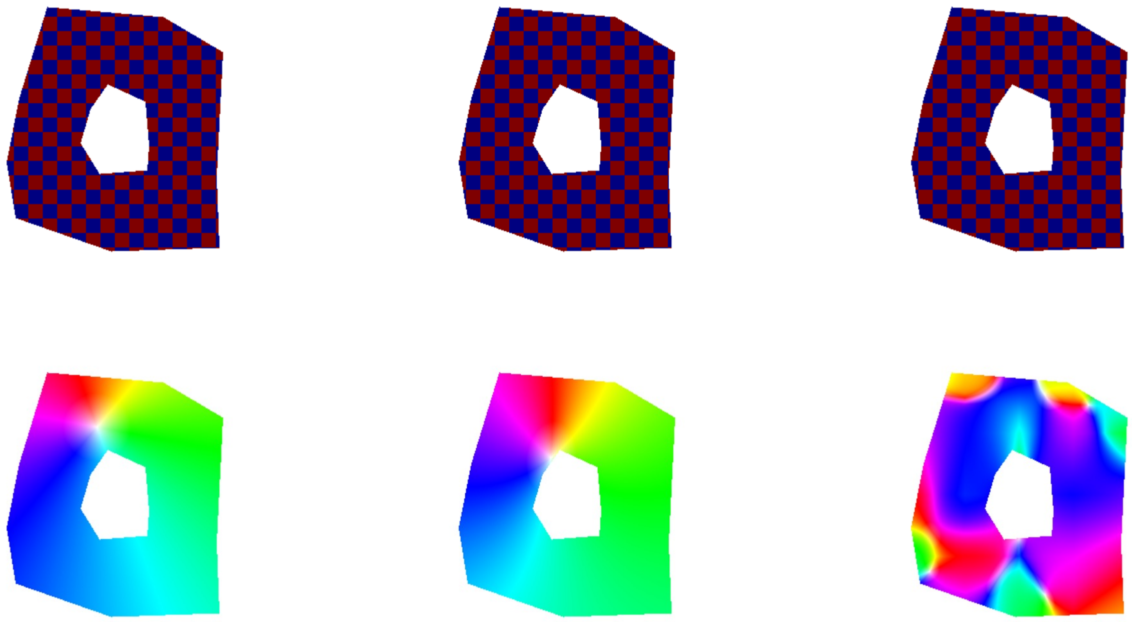
Modeling foreground shapes



Random polygons

Modeling motion





Affine

Perspective

Bilinear grid warp

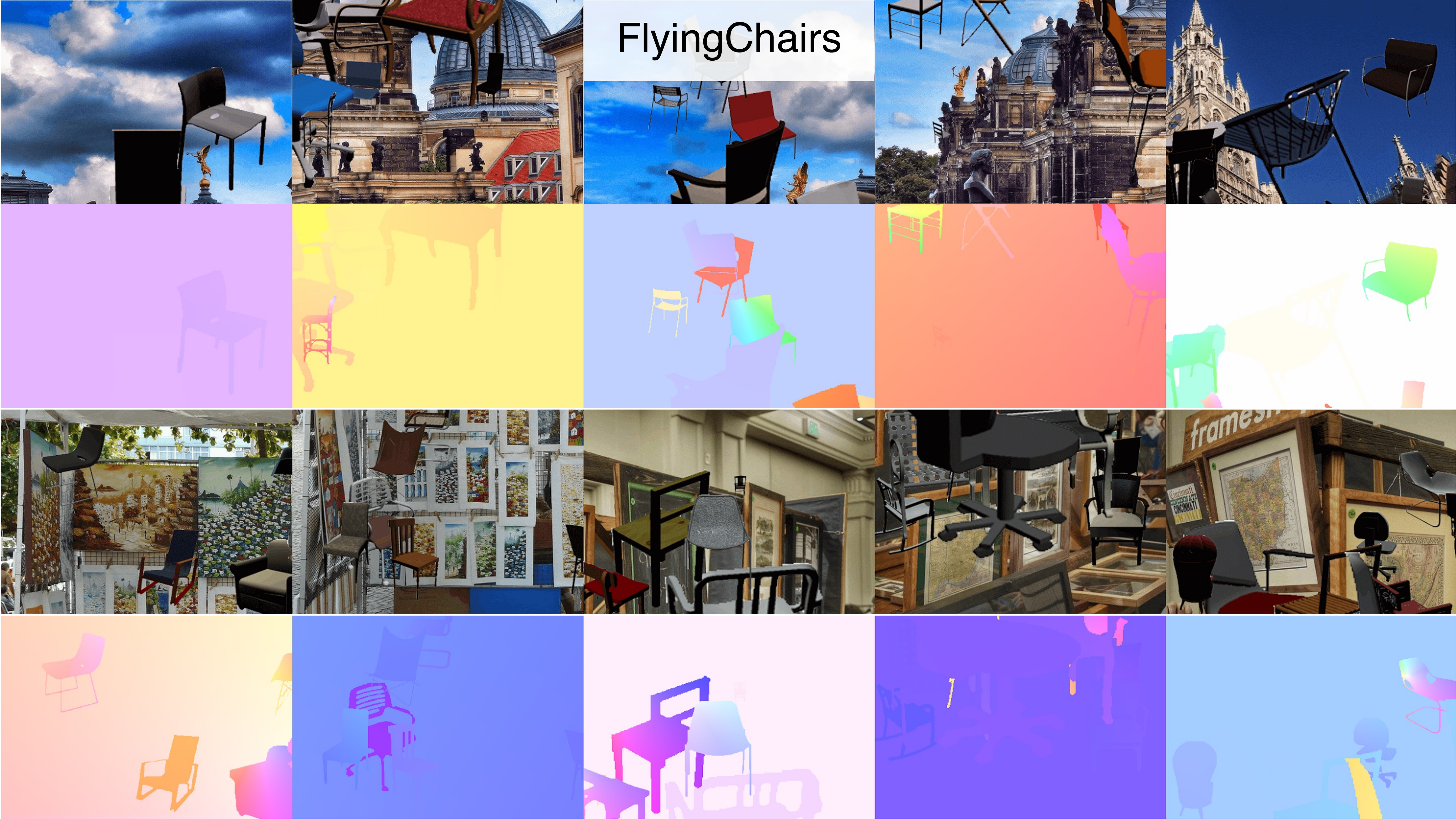
Visual effects

Motion blur

Fog


AutoFlow

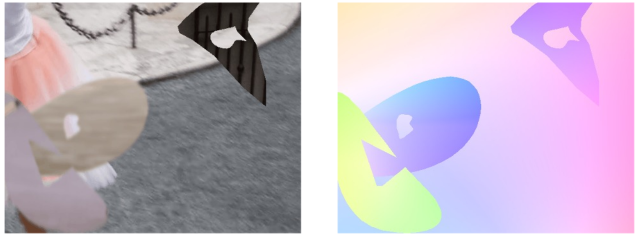
FlyingChairs



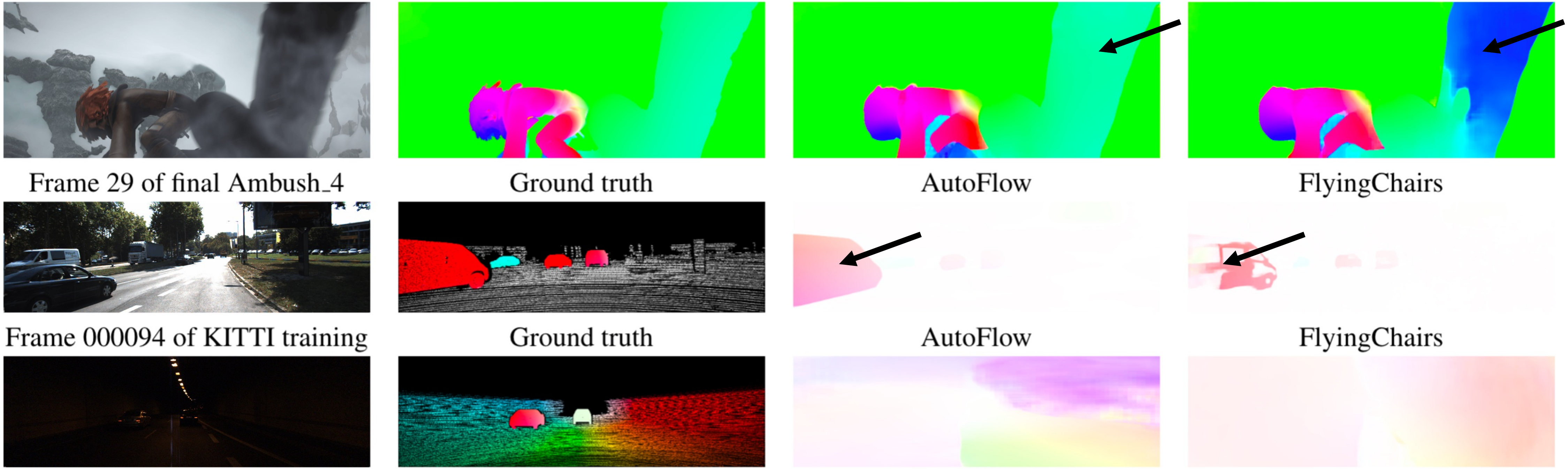
Results of pre-training (on training set)

Avg. end-point error (AEPE) ↓

Model	Dataset	Sintel.clean	Sintel.final	KITTI
PWC-Net	FlyingChairs	3.27	4.42	11.43
	Chairs→Things	2.39	3.90	9.81
	AutoFlow	**2.17**	**2.91**	**5.76**
RAFT	FlyingChairs	2.27	3.76	7.63
	Chairs→Things	**1.68**	2.80	5.92
	AutoFlow	1.95	**2.57**	**4.23**

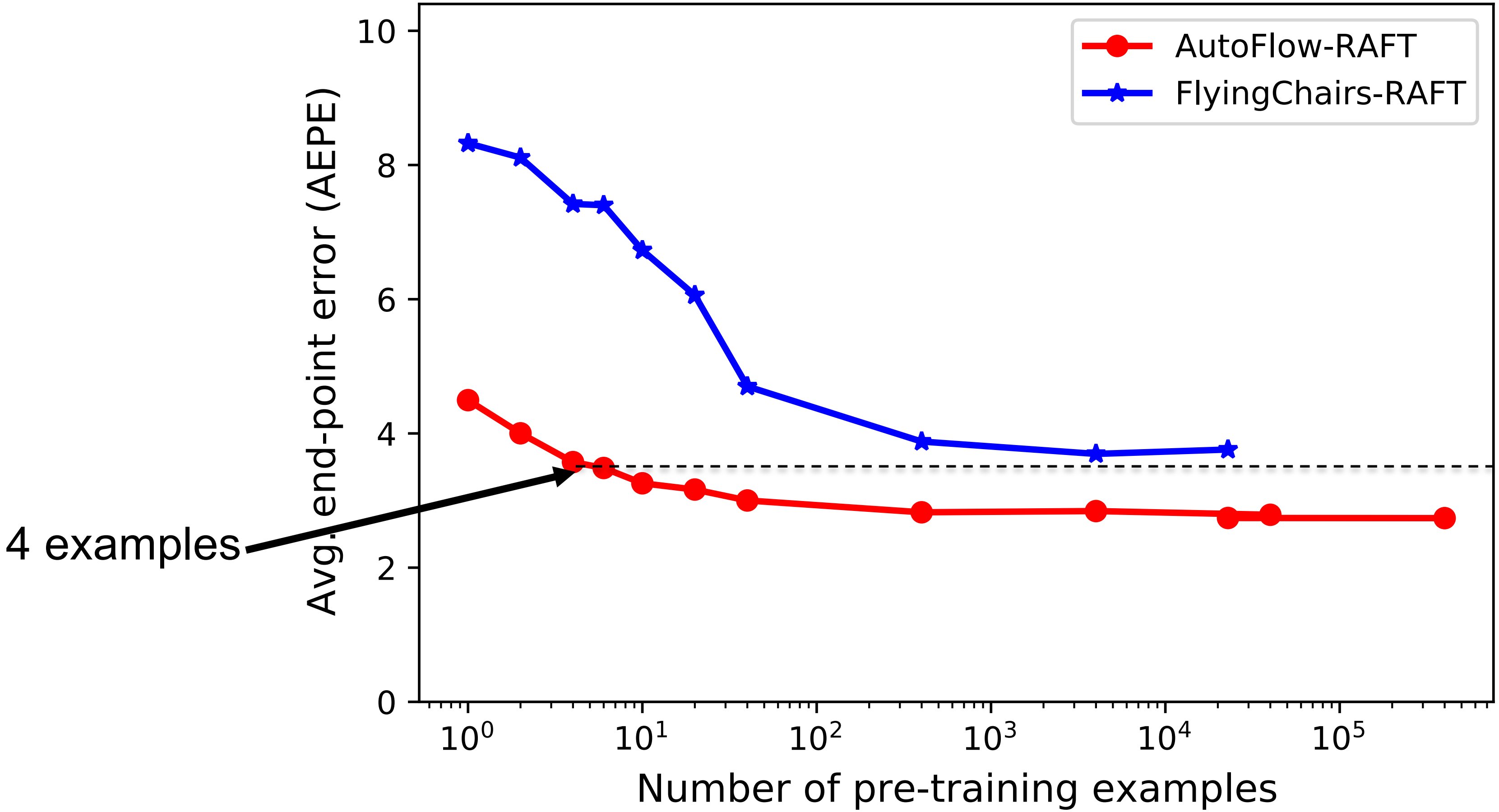


AutoFlow vs. FlyingChairs



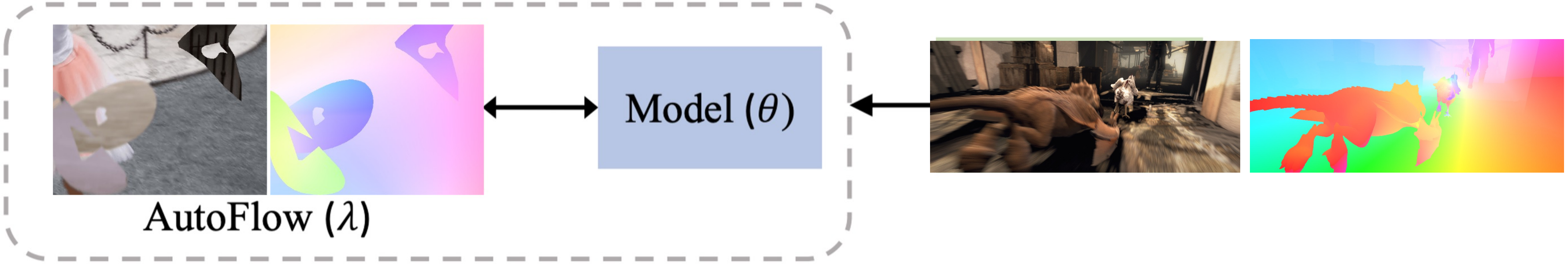
Frame 29 of final Ambush_4	Ground truth	AutoFlow	FlyingChairs
Frame 000094 of KITTI training	Ground truth	AutoFlow	FlyingChairs
Frame 000104 of KITTI training	Ground truth	AutoFlow	FlyingChairs

Number of training examples



Are architectures and data all we need?

Optimize the performance of a network on a target dataset



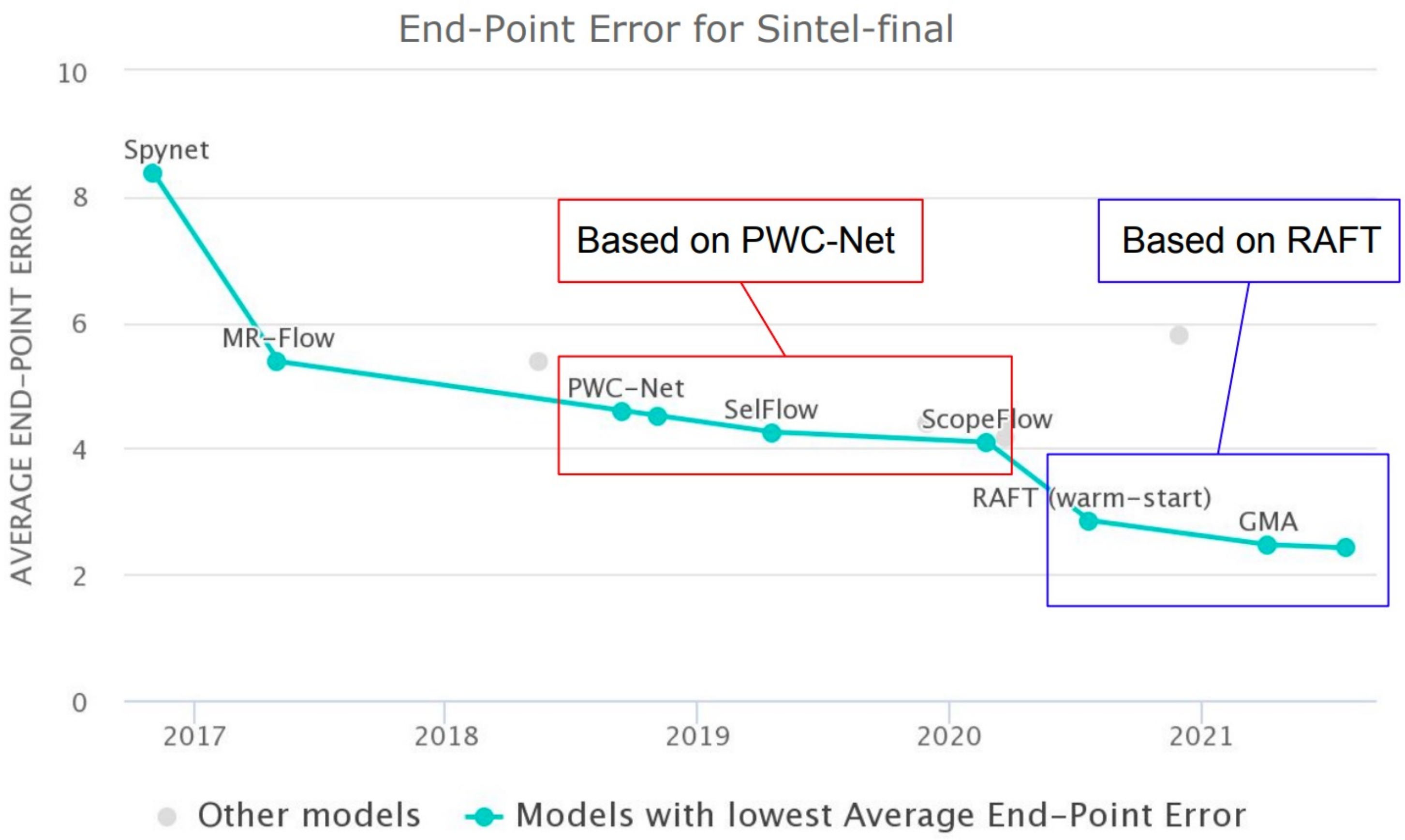
AutoFlow pipeline

Jointly render data and train model

Sun *et al.* AutoFlow: Learning a Better Training Set for Optical Flow, CVPR'21 Oral.

The devils are in the training details

Rapid progress on optical flow architectures

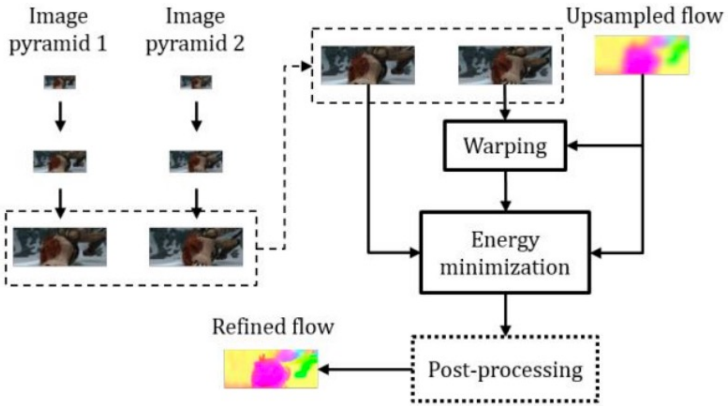


Differences in architecture

<u>PWC-Net (2018)</u>

Inspired by classical optical flow:

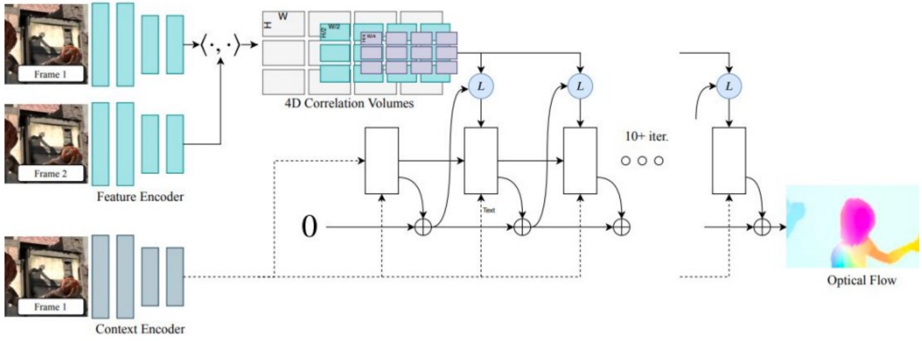
- Image pyramid
- Cost volume warping
- Multiscale loss function



<u>RAFT (2020)</u>

Introduced new network elements:

- Produces flow at a single level
- Multi-scale all-pairs cost volume
- Recurrent update operator
- Upsample module



Differences in training techniques

Lots of differences in training schemes

Training Details	PWC-Net (2018)	RAFT (2020)
Optimizer	Adam	AdamW
Learning rate schedule	Multi-Step LR	One Cycle LR
Gradient clipping	No	Yes
Training time	Moderate	Low
Augmentation	Color + Spatial	Different Color + Spatial

Imbalanced focus on architecture/modeling

Optical flow work on Modeling	Optical flow work on Training
Learning to estimate hidden motions with global motion aggregation. ICCV21	
High-resolution optical flow from 1d attention and correlation. ICCV21	
Separable flow: Learning motion cost volumes for optical flow estimation. ICCV21	*???*
Learning optical flow with adaptive graph reasoning. arxiv22	
Csflow: Learning optical flow via cross strip correlation for autonomous driving. arxiv22	
Pyramid recurrent all-pairs field transforms for optical flow estimation in robust vision challenge. arxiv22	

Disentangling architecture and training for optical flow

[Sun, Herrmann, *et al.* ECCV '22]

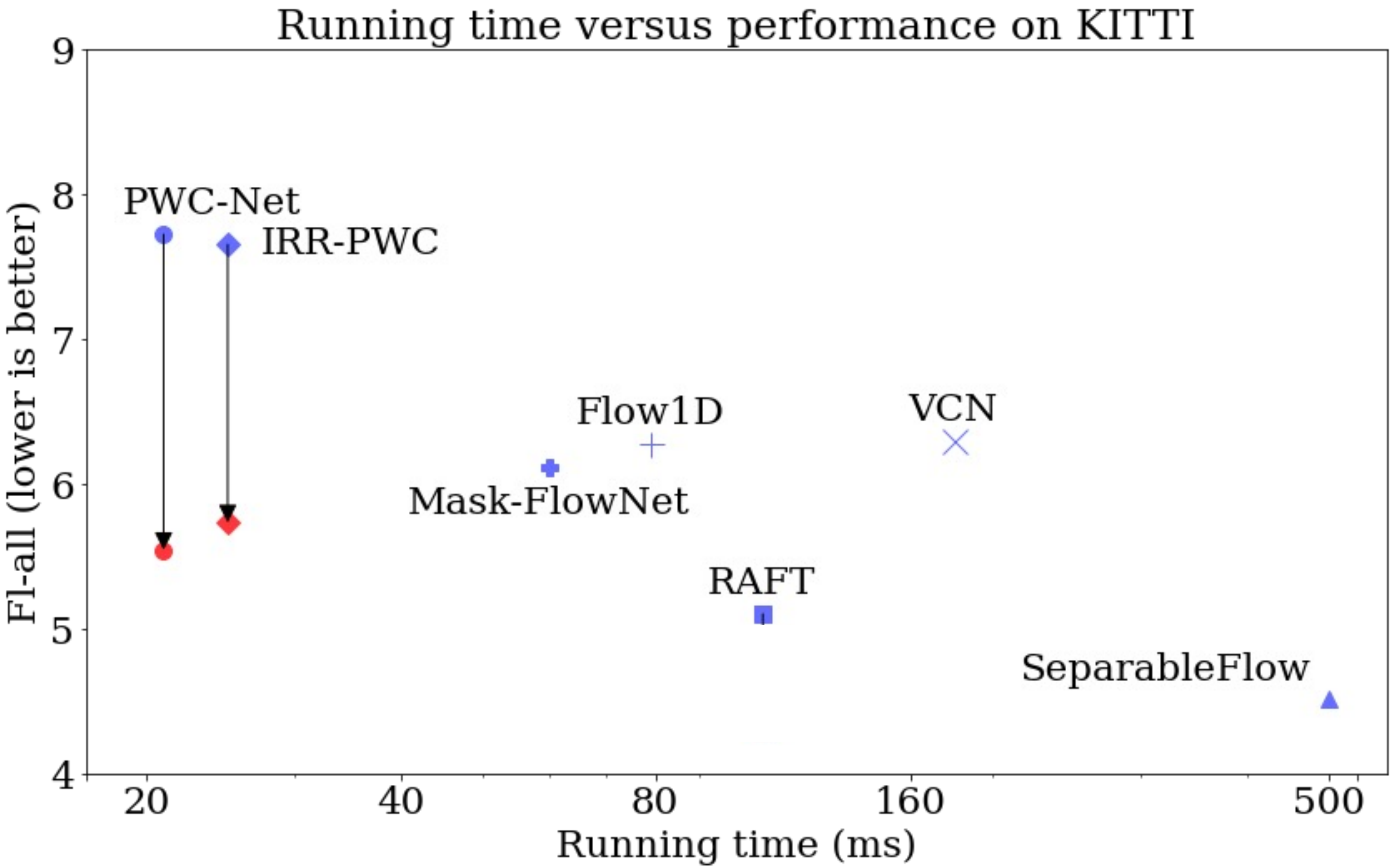
Our goals

- Understand the effect of modern training on performance and improve it.

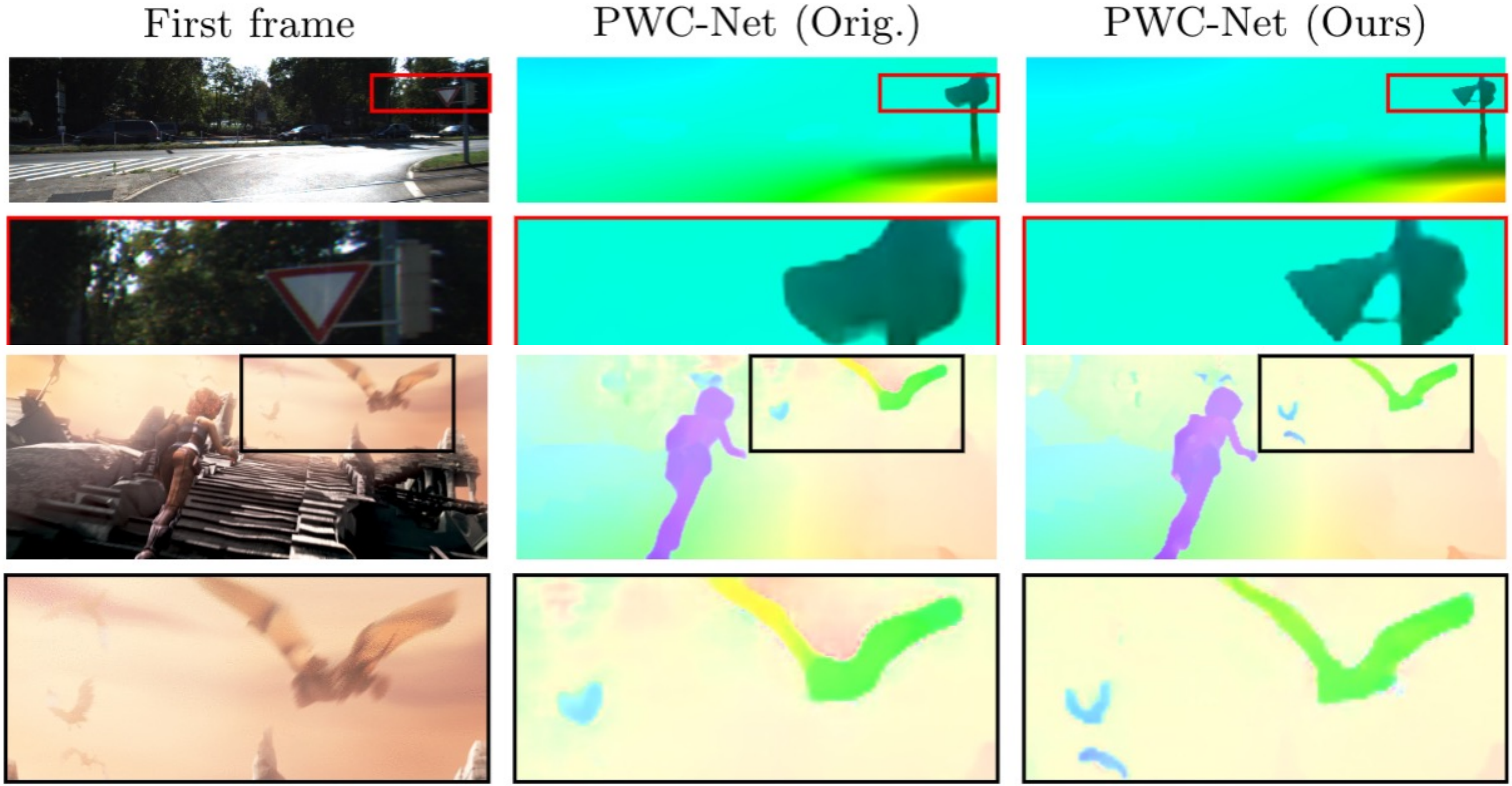
Our approach

- We apply a modern training scheme to 3 prominent models
  - PWC-Net (2018)
  - IRR-PWC (2019)
  - RAFT (2020)
- We perform a thorough ablation study on pre-training and fine-tuning

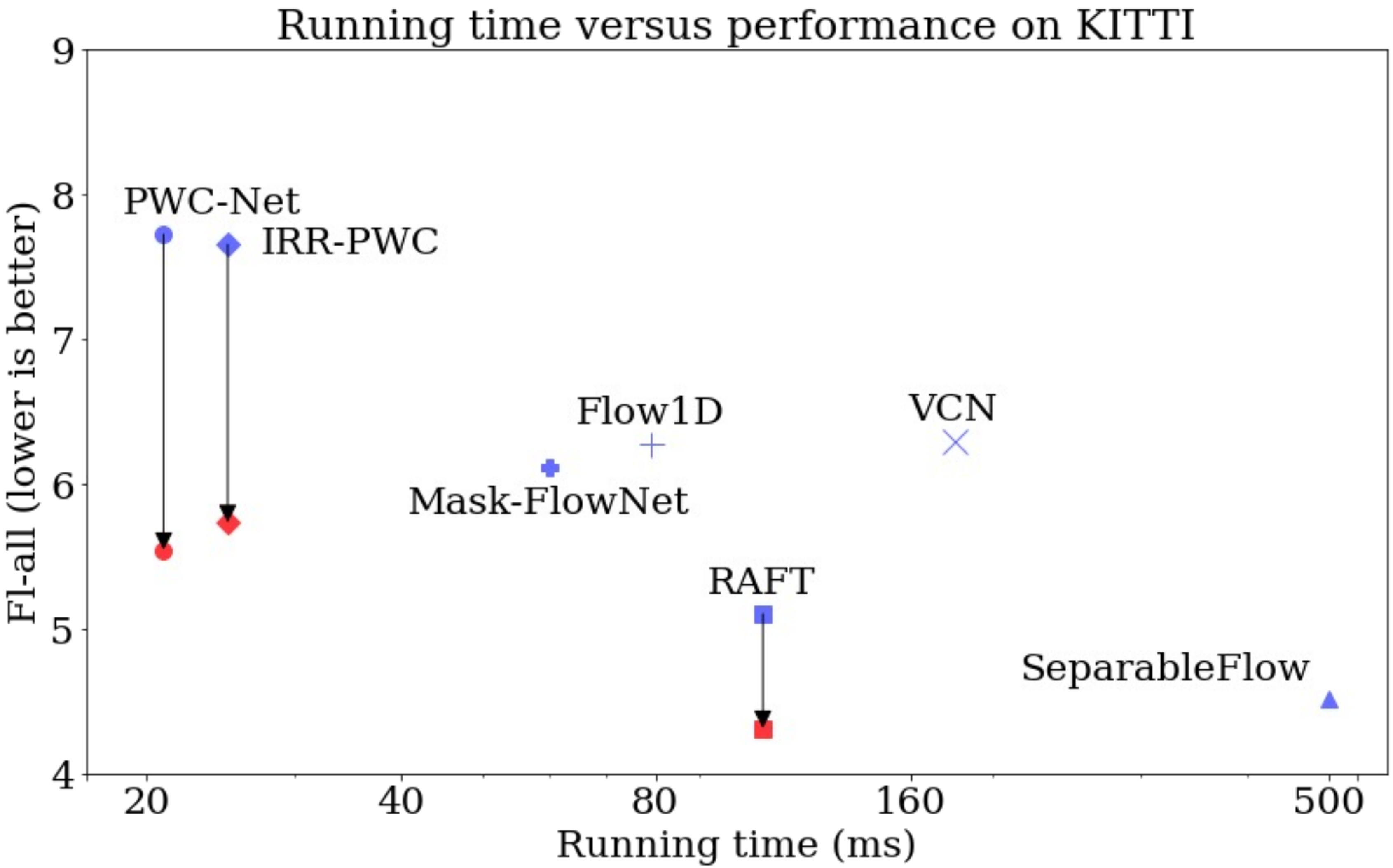
Better training significantly improves performance



“Old” vs. “new” PWC-Net



Better training significantly improves performance



Compared with state of the art (April 2023)

	Method	Setting	Code	Fl-bg	Fl-fg	Fl-all	Density	Runtime	Environment	Compare
1	CamLiRAFT		code	2.08 %	7.37 %	2.96 %	100.00 %	1 s	GPU @ 2.5 Ghz (Python + C/C++)	
H. Liu, T. Lu, Y. Xu, J. Liu and L. Wang: Learning Optical Flow and Scene Flow with Bidirectional Camera-LiDAR Fusion. arXiv preprint arXiv:2303.12017 2023.										
2	CamLiFlow		code	2.31 %	7.04 %	3.10 %	100.00 %	1.2 s	GPU @ 2.5 Ghz (Python + C/C++)	
H. Liu, T. Lu, Y. Xu, J. Liu, W. Li and L. Chen: CamLiFlow: Bidirectional Camera-LiDAR Fusion for Joint Optical Flow and Scene Flow Estimation. CVPR 2022.										
3	CamLiRAFT-NR		code	2.76 %	6.78 %	3.43 %	100.00 %	1 s	GPU @ 2.5 Ghz (Python + C/C++)	
H. Liu, T. Lu, Y. Xu, J. Liu and L. Wang: Learning Optical Flow and Scene Flow with Bidirectional Camera-LiDAR Fusion. arXiv preprint arXiv:2303.12017 2023.										
4	M-FUSE		code	2.56 %	7.47 %	3.46 %	100.00 %	1.3 s	GPU	
L. Mehl, A. Jahedi, J. Schmalfuss and A. Bruhn: M-FUSE: Multi-frame Fusion for Scene Flow Estimation. Proc. Winter Conference on Applications of Computer Vision (WACV) 2023.										
5	RigidMask+ISF		code	2.63 %	7.85 %	3.50 %	100.00 %	3.3 s	GPU @ 2.5 Ghz (Python)	
G. Yang and D. Ramanan: Learning to Segment Rigid Motions from Two Frames. CVPR 2021.										
6	ScaleRAFT3D			2.37 %	9.26 %	3.51 %	100.00 %	1 s	1 core @ 2.5 Ghz (C/C++)	
7	TPCV+RAFT3D			2.48 %	10.19 %	3.76 %	100.00 %	0.2 s	1 core @ 2.5 Ghz (C/C++)	
8	RAFT3D+mscv			2.48 %	10.23 %	3.77 %	100.00 %	0.2 s	1 core @ 2.5 Ghz (C/C++)	
9	RAFT-it+_RVC		code	3.62 %	5.33 %	3.90 %	100.00 %	0.14 s	1 core @ 2.5 Ghz (Python)	
D. Sun, C. Herrmann, F. Reda, M. Rubinstein, D. Fleet and W. Freeman: Disentangling Architecture and Training for Optical Flow. ECCV 2022.										
10	RAFT-OCTC			3.72 %	5.39 %	4.00 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	
J. Jeong, J. Lin, F. Porikli and N. Kwak: Imposing Consistency for Optical Flow Estimation (Qualcomm AI Research). CVPR 2022.										
11	RCA-Flow			3.67 %	6.25 %	4.10 %	100.00 %	0.16 s	1 core @ 2.5 Ghz (Python)	
12	SF2SE3		code	3.17 %	8.79 %	4.11 %	100.00 %	2.7 s	GPU @ >3.5 Ghz (Python)	
L. Sommer, P. Schröppel and T. Brox: SF2SE3: Clustering Scene Flow into SE (3)-Motions via Proposal and Selection. DAGM German Conference on Pattern Recognition 2022.										
13	RAFT-CF-CE-PL3			3.80 %	5.65 %	4.11 %	100.00 %	0.05 s	GPU @ 2.5 Ghz (Python)	

Using stereo input

What matters?

We iteratively build our training and show that each step improves Sintel.F

- *Dataset matters*

FlyingChairs ⇨ AutoFlow	for IRR	5.09 ⇨ 4.11	[-19%]
	for RAFT	4.03 ⇨ 3.36	[-17%]

- *Gradient clipping matters*

NoGC ⇨ YesGC	for IRR	4.11 ⇨ 3.29	[-20%]
	for RAFT	3.36 ⇨ 3.20	[-5%]

- *Training schedule matters*

Piecewise ⇨ OneCycle	for IRR	3.29 ⇨ 2.93	[-11%]
	for RAFT	3.20 ⇨ 2.75	[-14%]

- *Training iterations matter*

standard ⇨ 4-10 times more	for IRR	2.93 ⇨ 2.76	[-6%]
	for RAFT	2.75 ⇨ 2.41	[-12%]

Tradeoff between accuracy and speed/memory

	Inference Time (msec)↓			Peak Memory (GB)↓		
	1024×448	Full HD	4K	1024 × 448	Full HD	4K
PWC-Net	20.61	28.77	63.31	1.478	2.886	7.610
IRR-PWC	24.71	33.67	57.59	1.435	2.902	8.578
RAFT	107.38	499.63	n/a	2.551	9.673	OOM

Table 6. Inference time and memory usage for 1024×448, Full HD (1920×1080) and 4K (3840×2160) frame sizes, averaged over 100 runs on an NVIDIA V100 GPU.

Running on Full HD and 4K images

Inputs

IRR-PWC

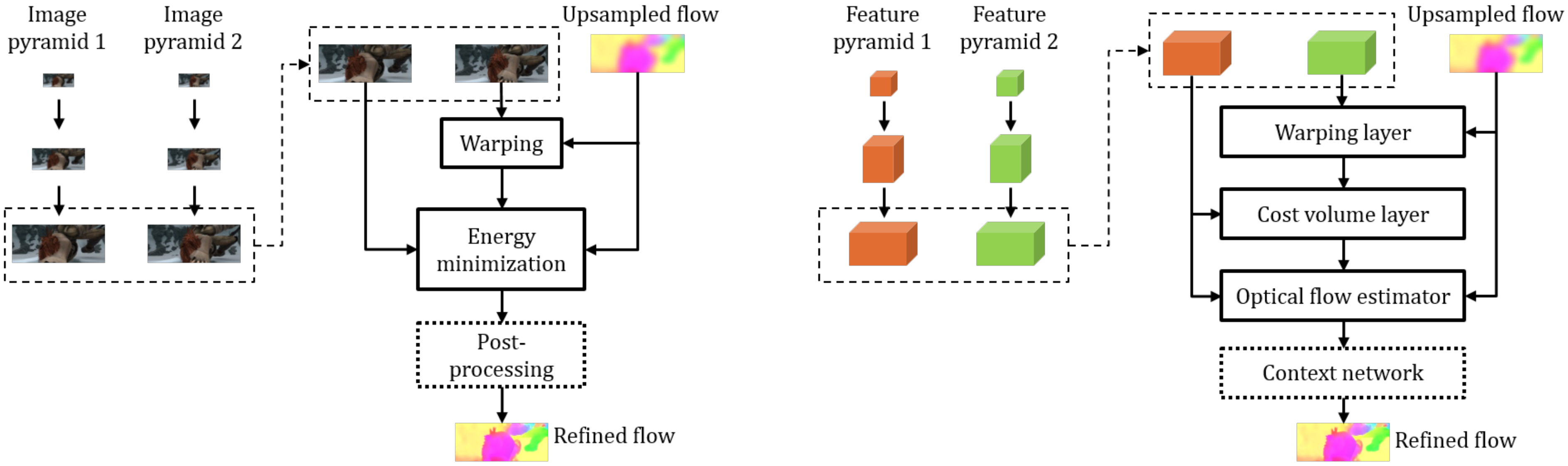
PWC-Net and IRR-PWC can run on 4K images, which cause RAFT to OOM


IRR-PWC on 4K input

RAFT on full HD input + x2 upsampling

Content

- Deep learning-based approach
  - Designing architecture (using domain knowledge)



Content

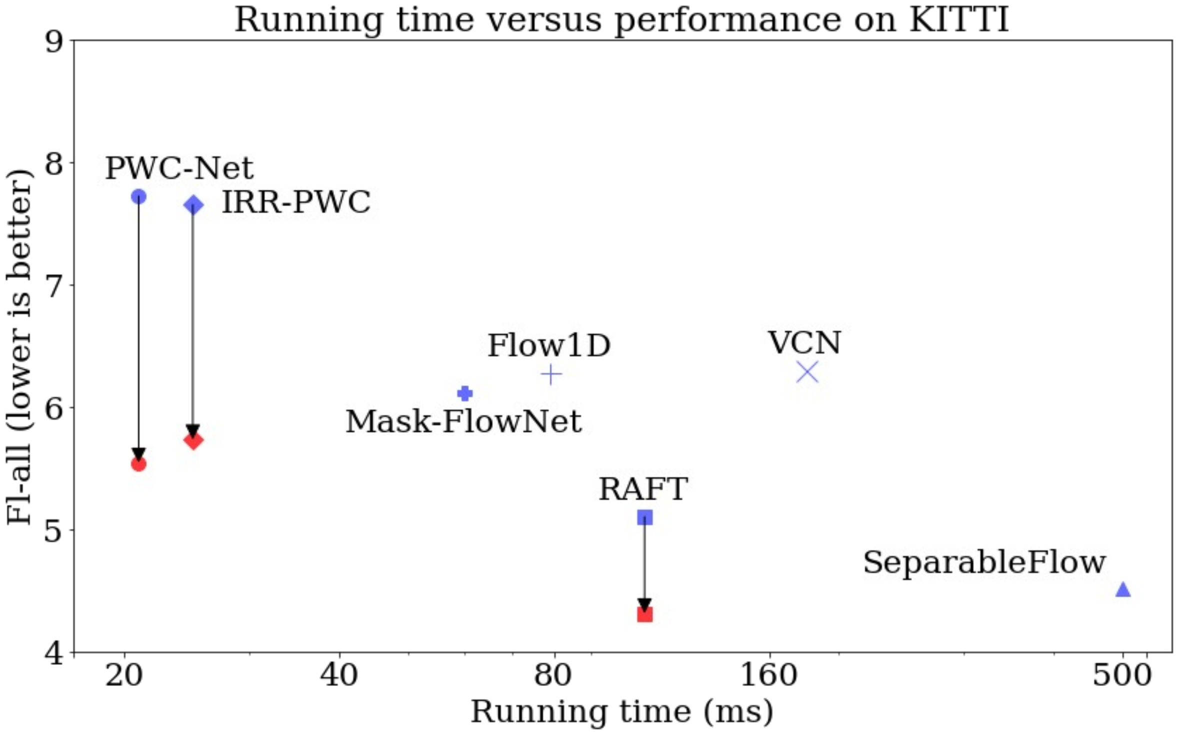
- Deep learning-based approach
  - Designing architecture (using domain knowledge)
  - Learning data (matters)

FlyingChairs: Manually designed

AutoFlow: Joint data generation and network training

Content

- Deep learning-based approach
  - Designing architecture (using domain knowledge)
  - Learning data (matters)
  - Evaluating architectures fairly (trade-off in accuracy and speed/memory)



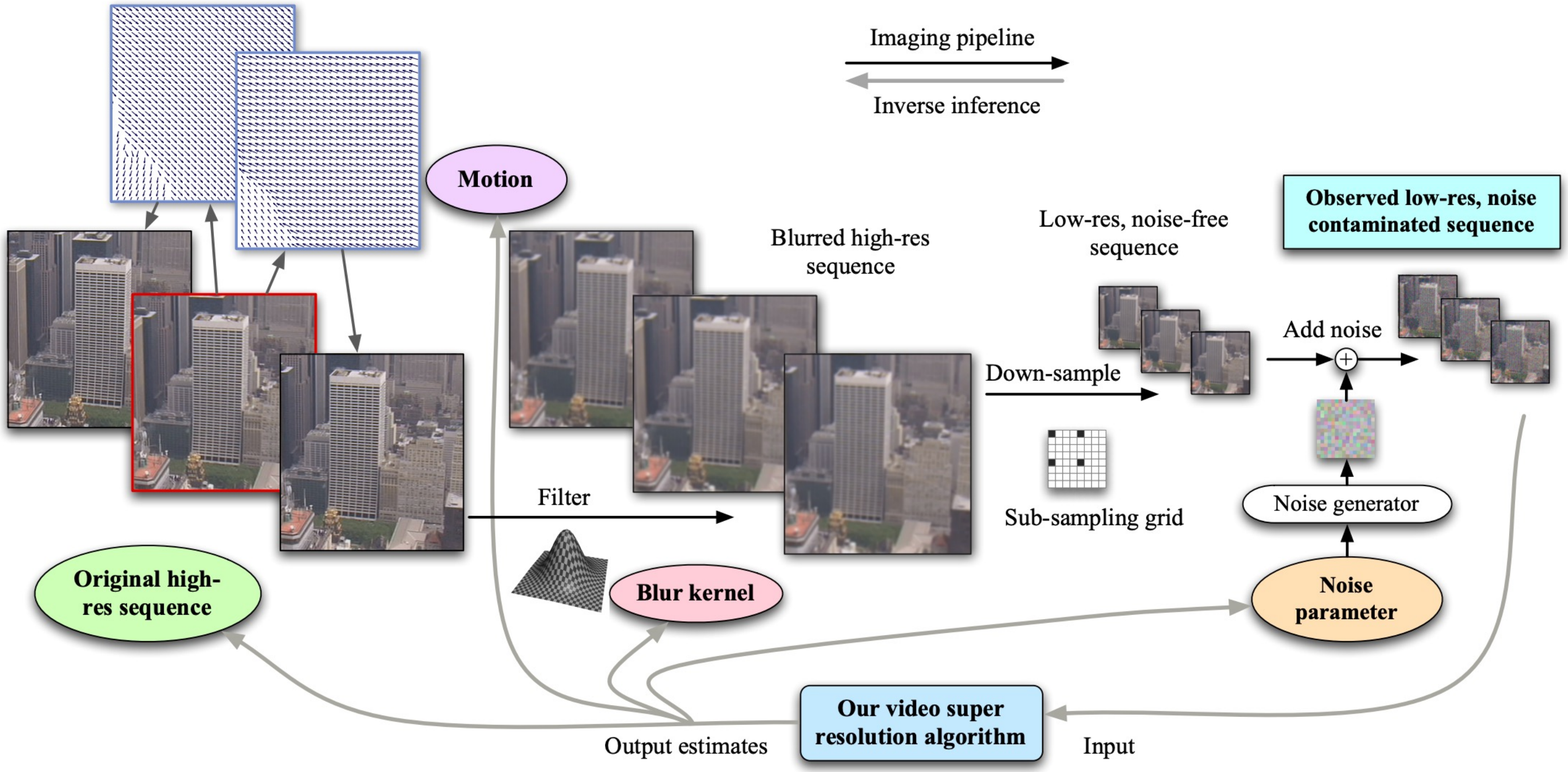
Results on real-world videos

What is motion for?

Video super-resolution

[Liu & Sun CVPR 2011, TPAMI 2014]

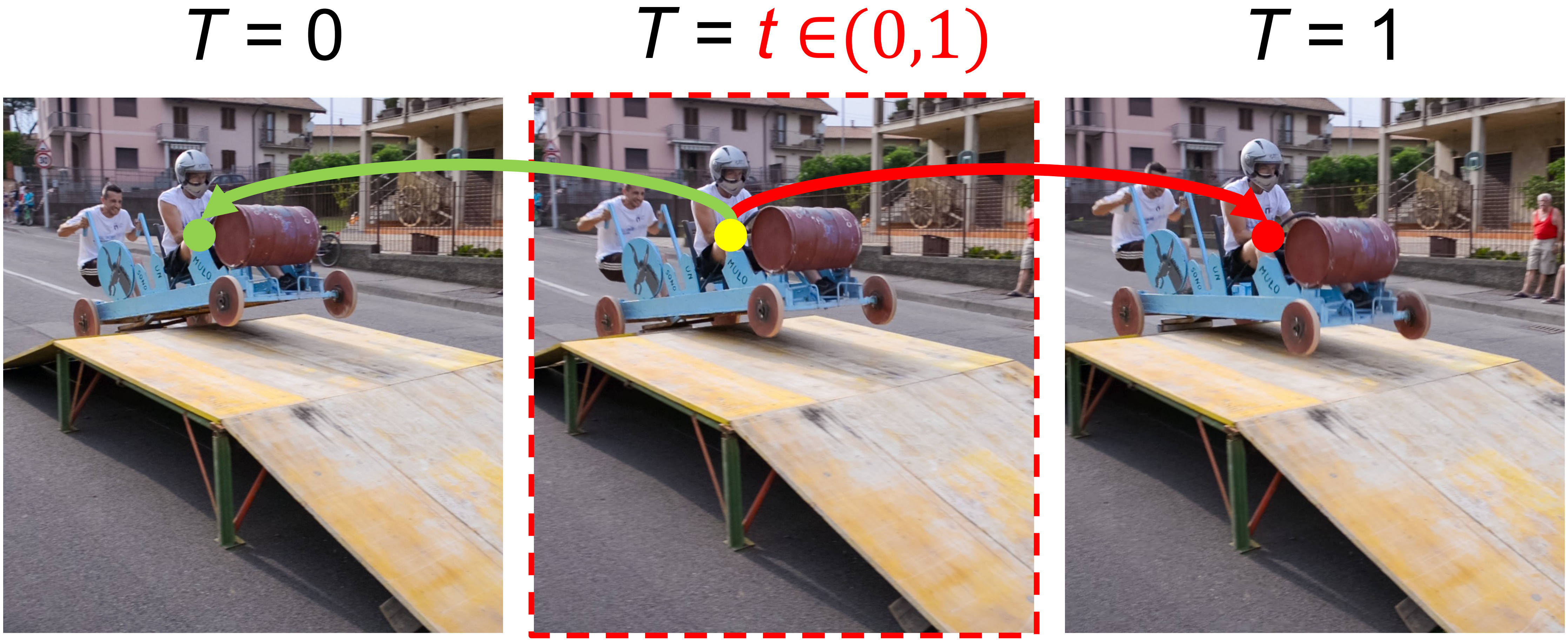
Video super-resolution



Video frame interpolation

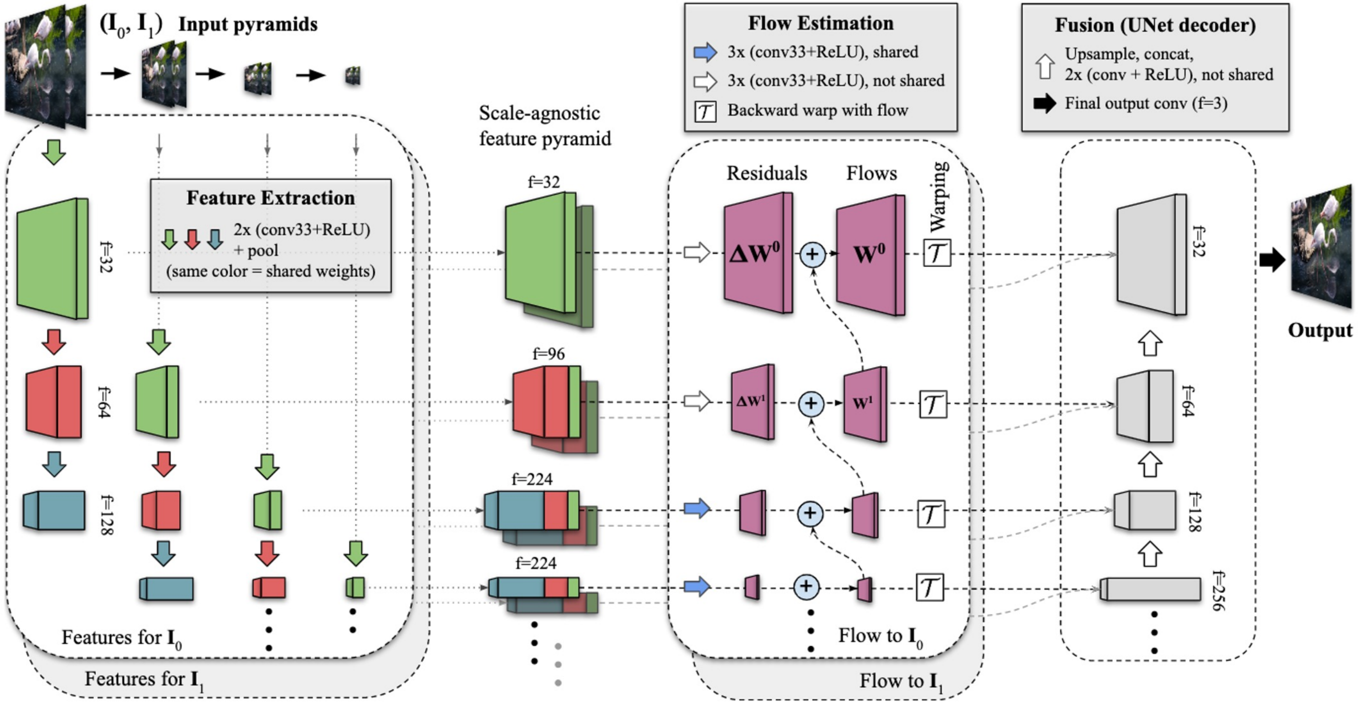
Super SloMo [Jiang, Sun, *et al.* CVPR 2018]
Incorporated into **NVIDIA NGX** SDK for the Turing GPU.

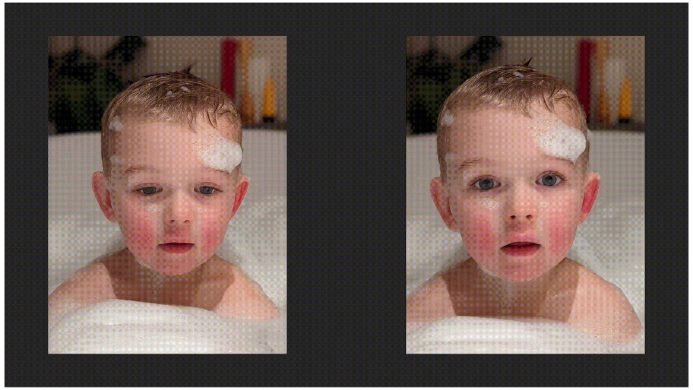
Idea: frame synthesis using optical flow



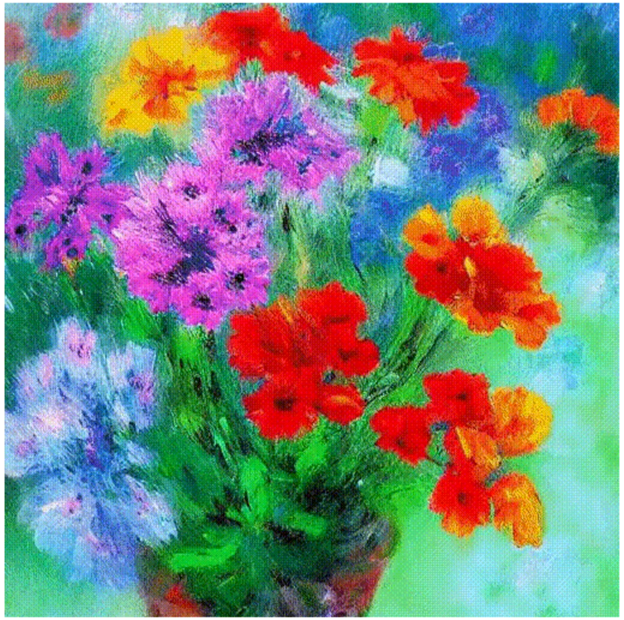
FILM: Frame Interpolation for Large Motion

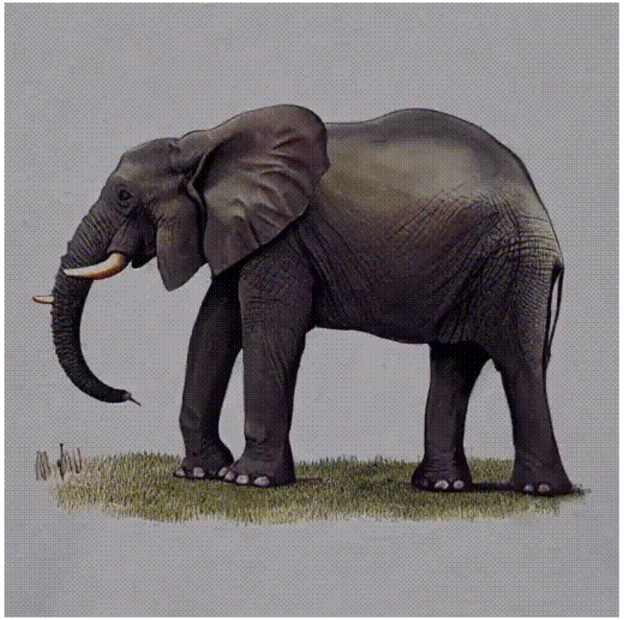
[Reda, *et al.* ECCV 2022]





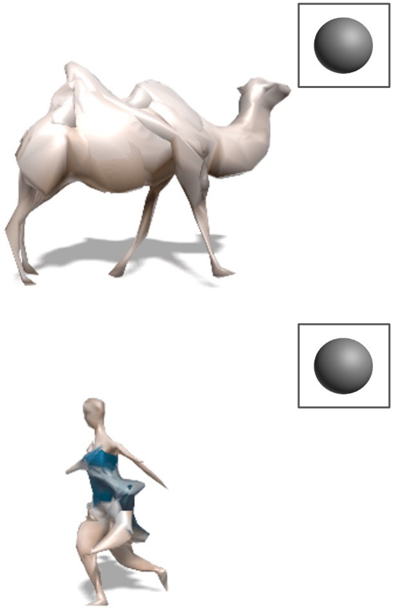
Stable diffusion + FILM (from twitter)

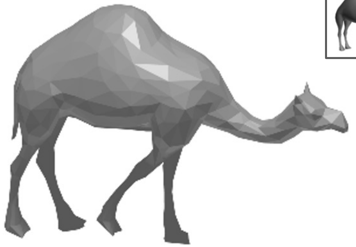




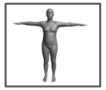
LASR: Learning Articulated Shape Reconstruction from a monocular video

[Yang, *et al.* CVPR 2021]





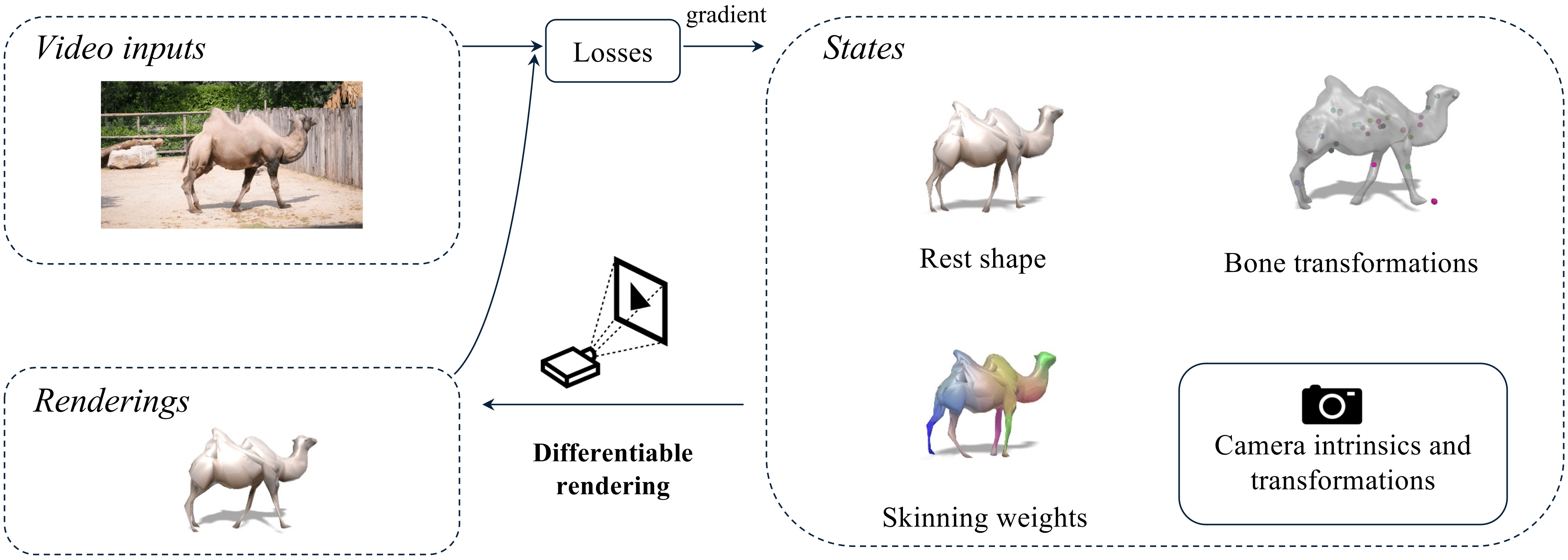
A-CSM: Kulkarni *et al.* CVPR 2020



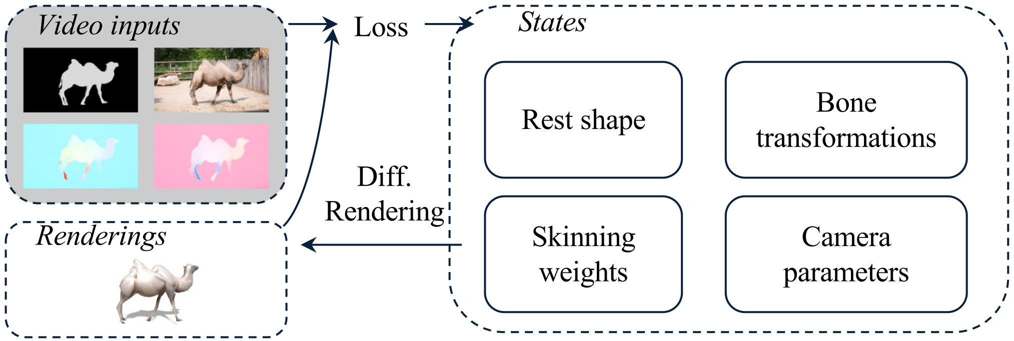
VIBE: Kocabas *et al.* CVPR 2020

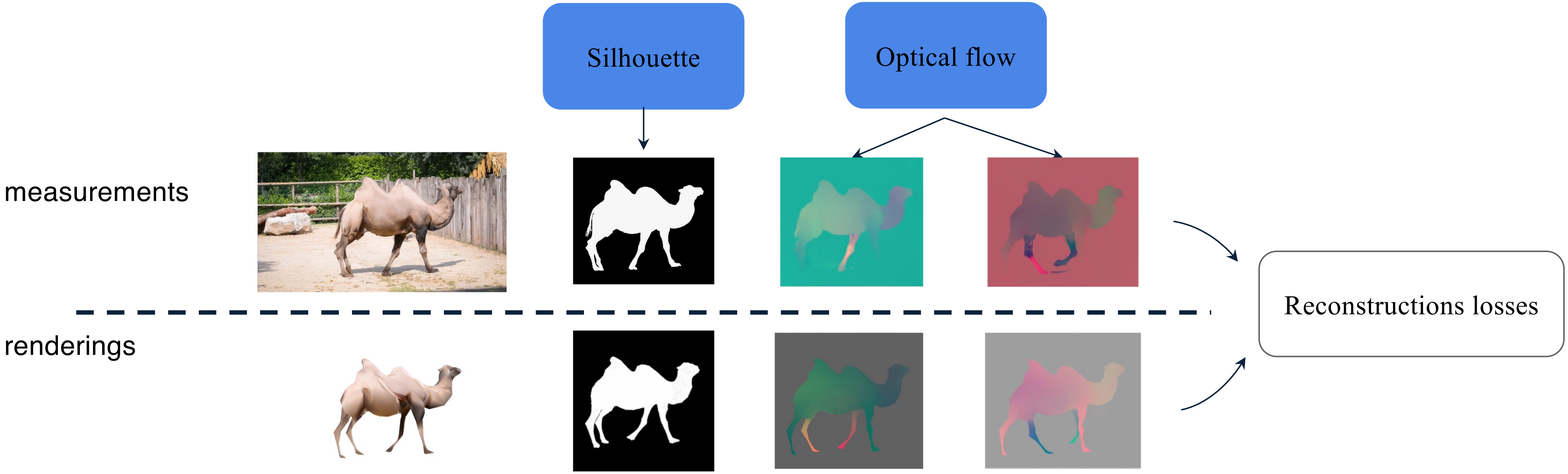
Challenge: Solving non-rigid 3D shape from 2D measurements **without template or category prior** is highly *under-constrained*

Approach: Analysis-by-synthesis

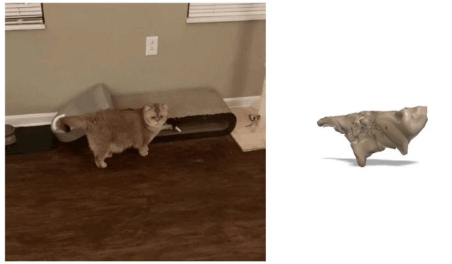


Supervision from silhouette, flow and pixels





Reconstructions on more real videos



Input Reconstructions Input Reconstructions Input Reconstructions

Face Unblur for Pixel 6

[Lai *et al.* SIGGRAPH 2022]

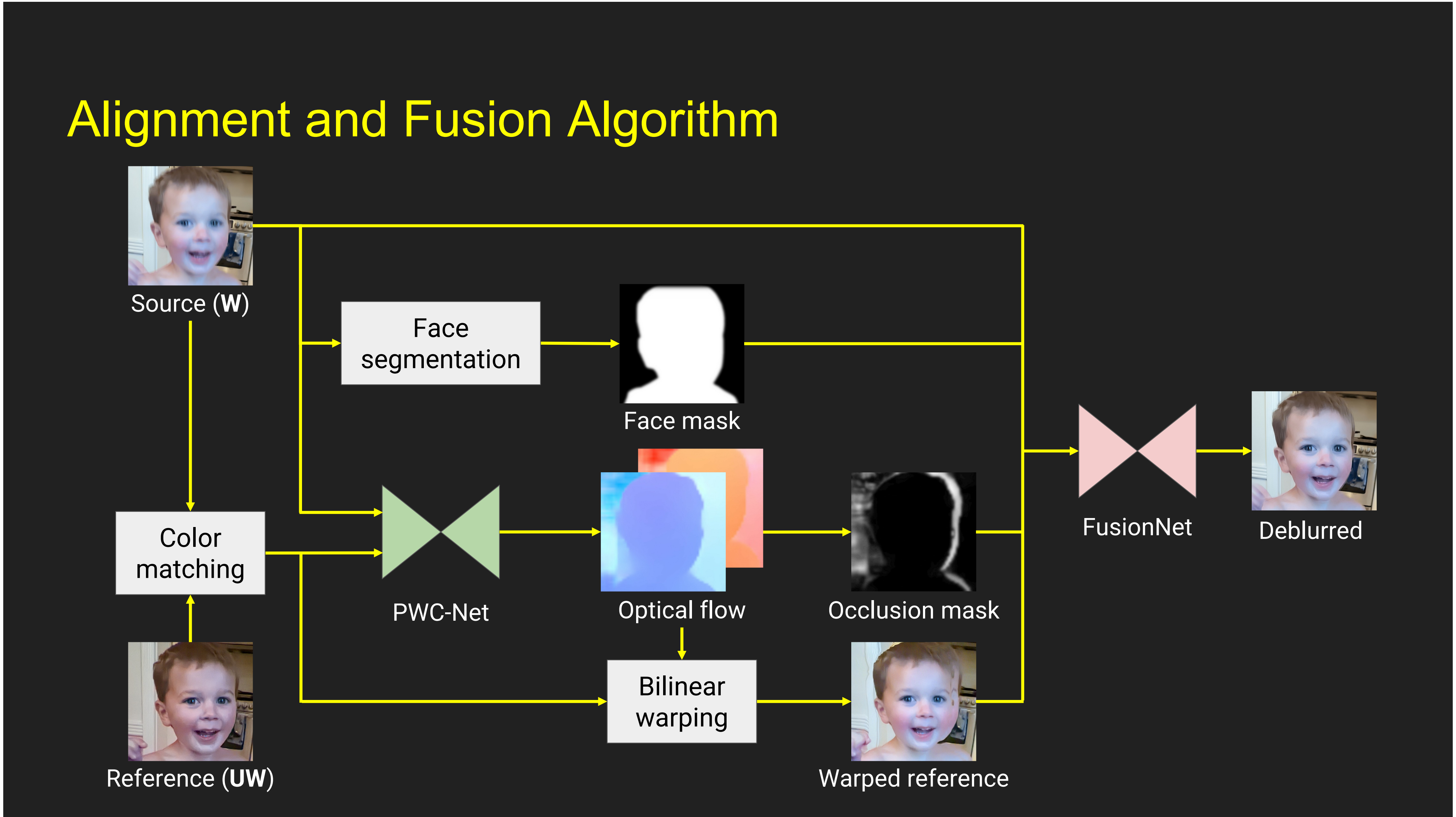
Google Pixel 6 Pro

Key Idea: Wide + Ultrawide Dual Camera Fusion

Wide (1/120 s)

UW (1/480 s)

Face Unblur



Alignment and Fusion Algorithm
Source (W)
Face segmentation
Face mask
Color matching
PWC-Net
Optical flow
Occlusion mask
FusionNet
Deblurred
Bilinear warping
Reference (UW)
Warped reference

Real-time optical flow on Pixel 6

Input image pairs

Flow by unoptimized model (>9000ms, 2GB memory)

Flow by optimized model (~**13ms, 34MB** memory)

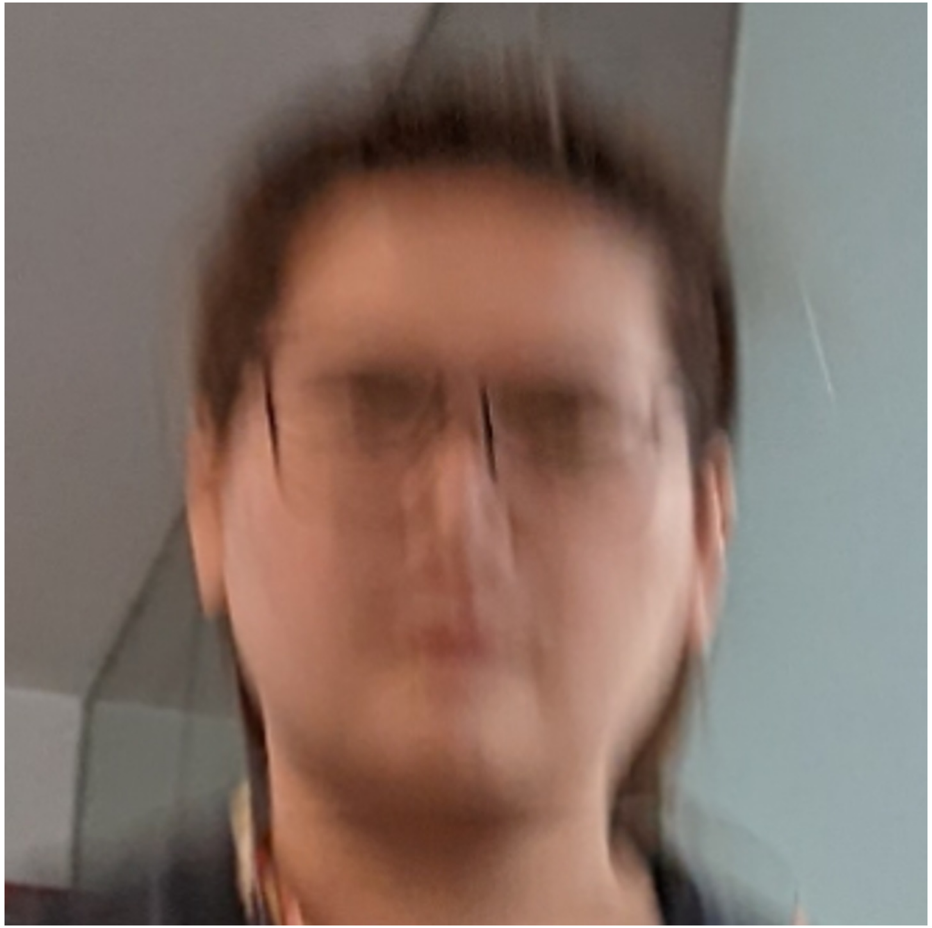
Input: Kids Standing Up

Our Deblurred Result

Input: Dynamic Motion

Our Deblurred Result

Input: Walking



Our Deblurred Result

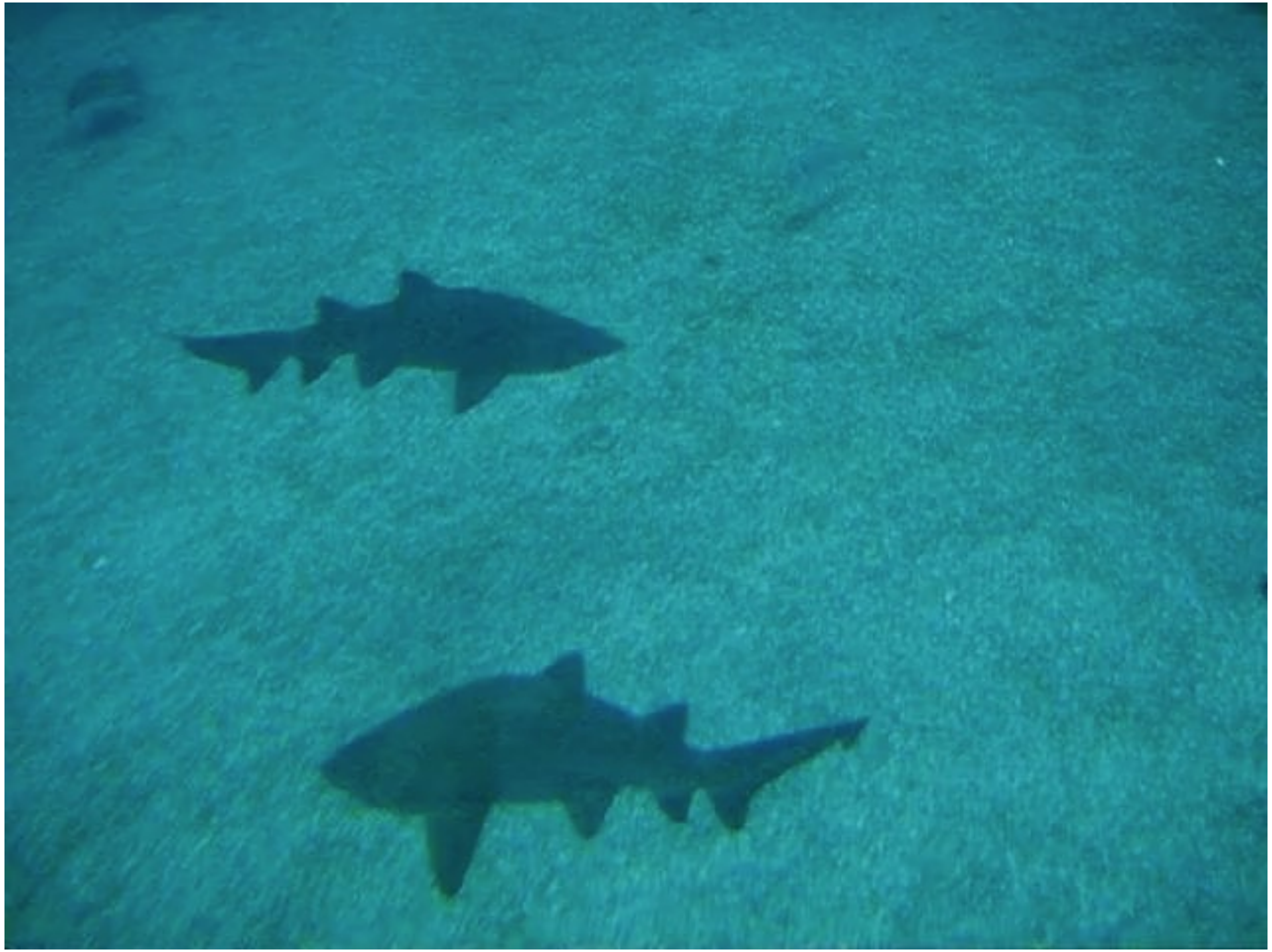
What we haven’t covered

Multiple motions

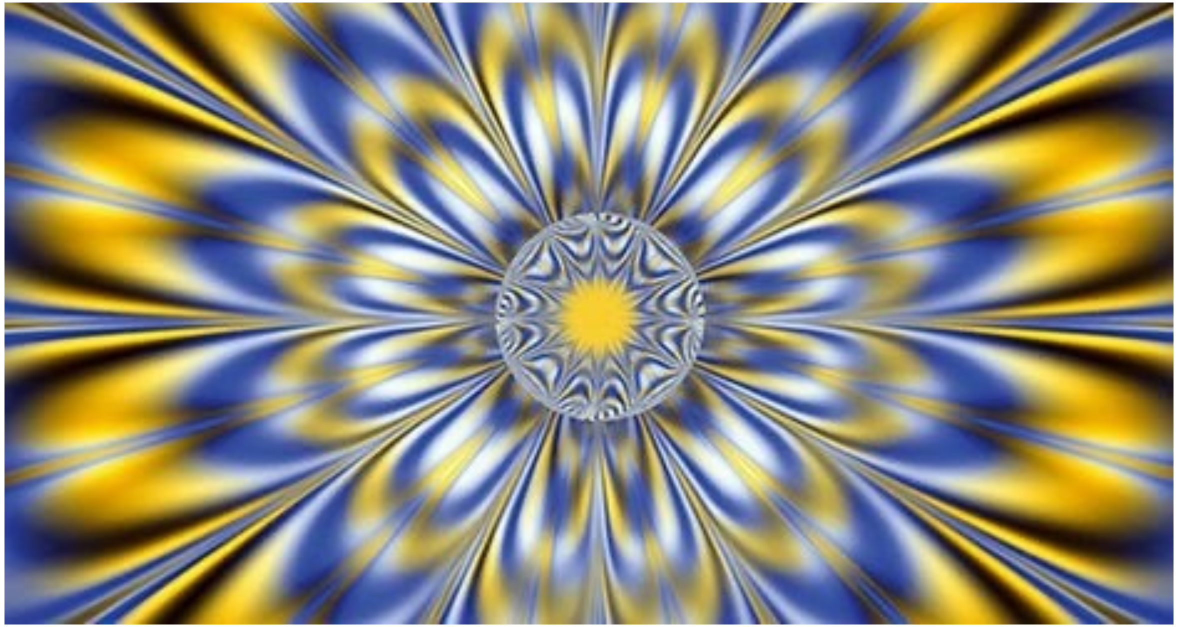
Fine details

Even harder

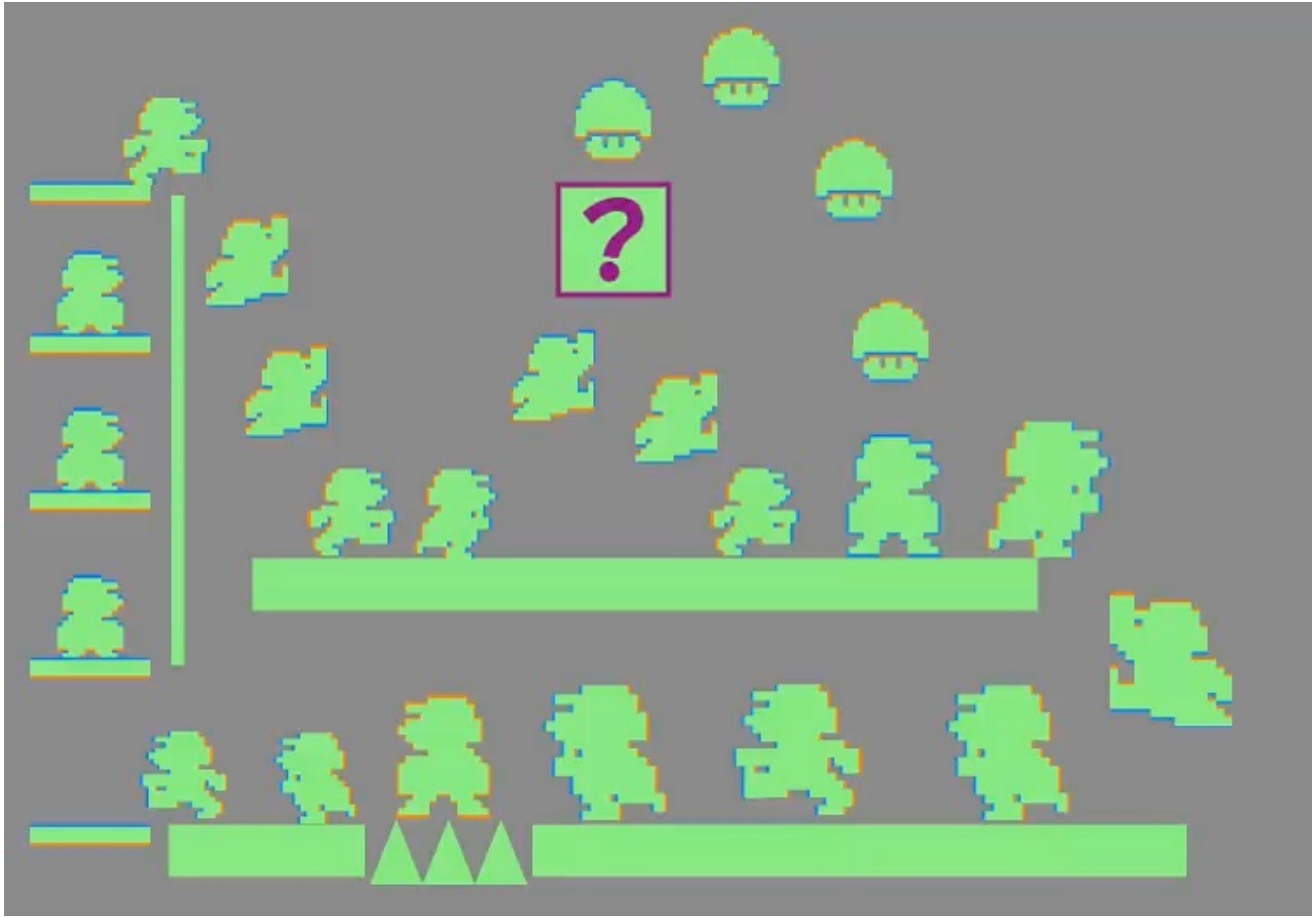
How to obtain ground truth for real-world videos?



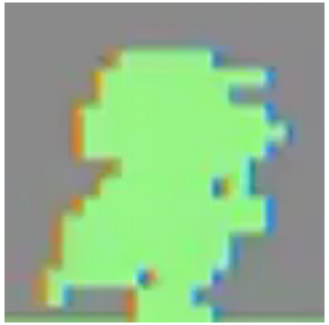
How do humans perceive motion?



What do you perceive?

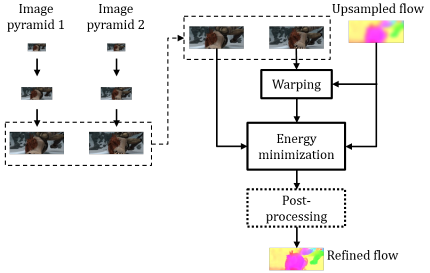


A single Mario

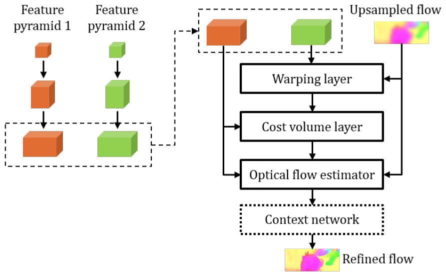


Content

- Classical approach
  - Constancy assumption -> matching by comparison (cost volume)
  - Coarse-to-fine, warping-based iterative estimation



- Deep learning-based approach
  - Designing architecture (using domain knowledge)
  - Learning data (matters)
  - Evaluating architectures fairly (trade-off in accuracy and speed/memory)



- Applications: What is motion for?
  - Super-resolution, frame interpolation, articulated 3D reconstruction …
  - Face Unblur (real-time dense accurate flow on mobile device)

A “biased” reading list

- Dr. Rick Szeliski’s book (2nd edition) chapter 9 on motion estimation
- Chapters 40-43 of book by Antonio, Phillip and Bill
- Horn & Schunck, Lucas & Kanade, Secrets of optical flow
- FlowNet, PWC-Net, IRR-PWC, RAFT, Perceiver IO, GM-Flow, FlowFormer(++), AutoFlow, Disentangling architecture and training
- Bayesian VSR, Super SloMo, FILM, LASR, Face Unblur

Thank you!

Deqing Sun
deqingsun@google.com