RAFT: Recurrent All-Pairs Field Transforms for Optical Flow

Zachary Teed and Jia Deng

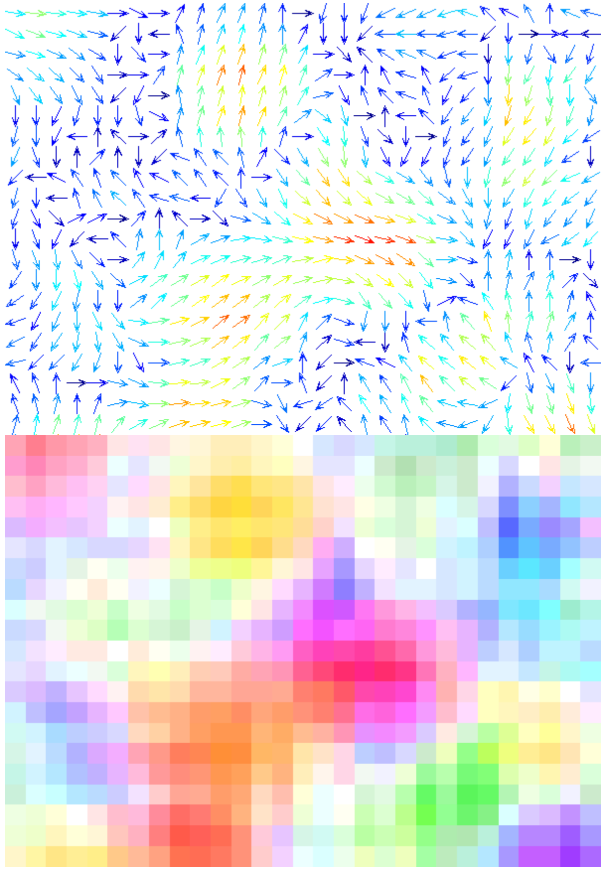
Princeton University
{zteed,jiadeng}@cs.princeton.edu

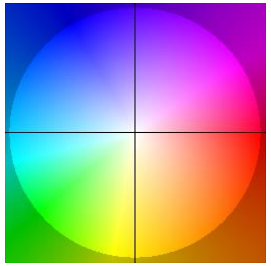
ECCV 2020 BEST PAPER AWARD

2020. 07. 29

Jinhee Kim

1 Optical flow





(direction , magnitude)

The optical flow field color-coding. Smaller vectors are lighter and color represents the direction.

Sintel light field video dataset

Bamboo clean

Temple clean

Bamboo final

Temple final

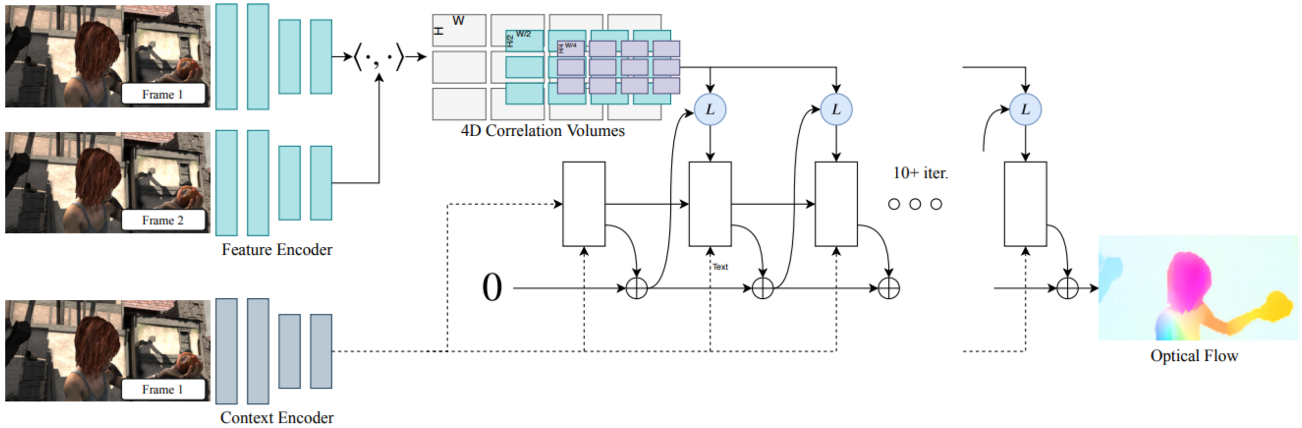
Optical flow

Optical flow

2 Overview

RAFT (end-to-end)

1. Feature extraction
2. Visual similarity computation
3. Iterative updates



Contribution

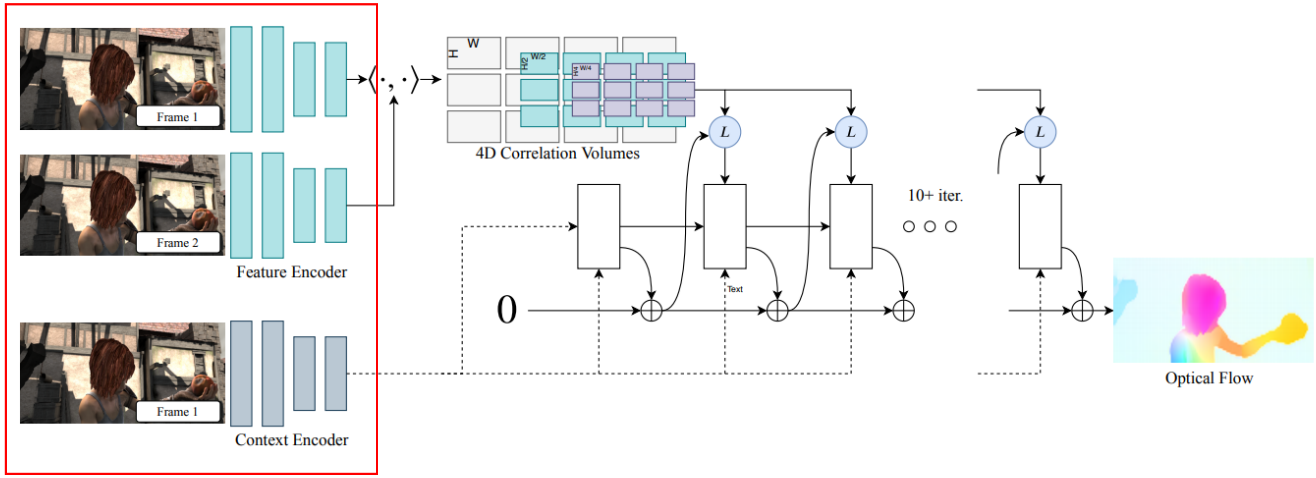
- RAFT **maintains and updates a single fixed flow field at high resolution** unlike the prevailing coarse-to-fine design.
  - **limitations of a coarse-to-fine cascade**
    - difficulty of recovering from errors at coarse resolutions
    - tendency to miss small fast-moving objects
    - many training iterations
- **the update operator** of RAFT is
  - recurrent and lightweight: **only 2.7M parameters and can be applied 100+ times during inference without divergence**
  - novel: it consists of **a convolutional GRU that performs lookups on 4D multi-scale correlation volumes**
- RAFT achieves **state-of-the-art performance on Sintel and KITTI datasets**

1. **Feature extraction**

$$g_\theta : \mathbb{R}^{H \times W \times 3} \mapsto \mathbb{R}^{H/8 \times W/8 \times D}$$

where we set $D = 256$.

h_θ : identical to the feature extraction network



2. Visual similarity computation

Given image features $g_\theta(I_1) \in \mathbb{R}^{H \times W \times D}$ and $g_\theta(I_2) \in \mathbb{R}^{H \times W \times D}$
correlation volume is formed by taking the dot product between all pairs of feature vectors. The correlation volume, $\mathbf{C}$, can be efficiently computed as a single matrix multiplication.

$$\mathbf{C}(g_\theta(I_1), g_\theta(I_2)) \in \mathbb{R}^{H \times W \times H \times W}, \qquad C_{ijkl} = \sum_h g_\theta(I_1)_{ijh} \cdot g_\theta(I_2)_{klh}$$

Correlation Pyramid: a 4-layer pyramid $\{\mathbf{C}^1, \mathbf{C}^2, \mathbf{C}^3, \mathbf{C}^4\}$
volume $\mathbf{C}^k$ has dimensions $H \times W \times H/2^k \times W/2^k$

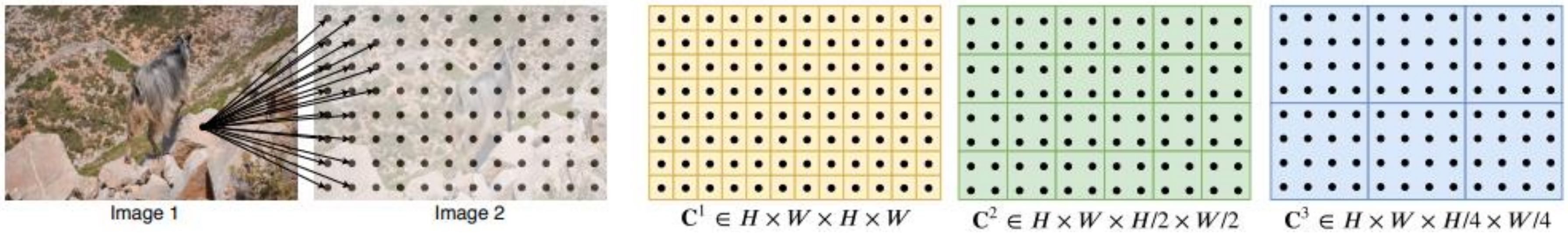


Fig. 2: Building correlation volumes. Here we depict 2D slices of a full 4D volume. For a feature vector in I_1, we take take the inner product with all pairs in I_2, generating a 4D $W \times H \times W \times H$ volume (each pixel in I_2 produces a 2D response map). The volume is pooled using average pooling with kernel sizes $\{1, 2, 4, 8\}$.

2. Visual similarity computation

Correlation Lookup: generates a feature map by indexing from the correlation pyramid.

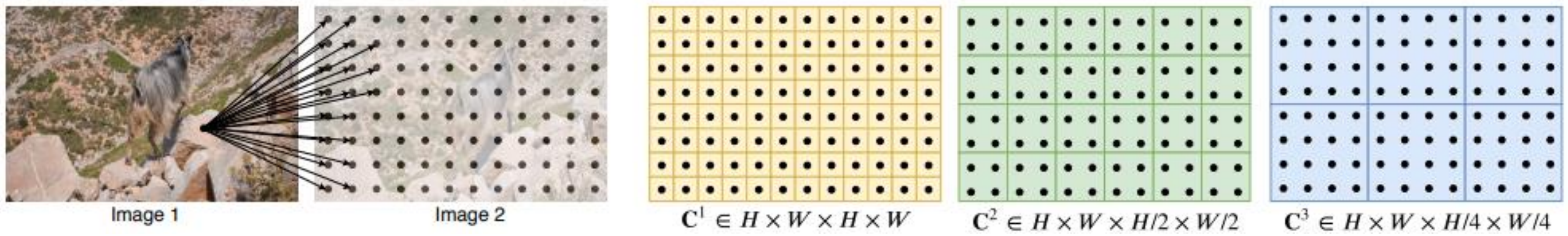
optical flow $(\mathbf{f}^1, \mathbf{f}^2)$

we map each pixel $\mathbf{x} = (u, v)$ in I_1 to its estimated correspondence in I_2: $\mathbf{x}' = (u + f^1(u), v + f^2(v))$. We then define a local grid around $\mathbf{x}'$

$$\mathcal{N}(\mathbf{x}')_r = \{\mathbf{x}' + \mathbf{dx} \mid \mathbf{dx} \in \mathbb{Z}^2, ||\mathbf{dx}||_1 \leq r\} \tag{2}$$

We use the local neighborhood $\mathcal{N}(\mathbf{x}')_r$ to index from the correlation volume. Since $\mathcal{N}(\mathbf{x}')_r$ is a grid of real numbers, we use bilinear sampling.

We perform lookups on all levels of the pyramid, such that the correlation volume at level k, $\mathbf{C}^k$, is indexed using the grid $\mathcal{N}(\mathbf{x}'/2^k)_r$. A constant radius across levels means larger context at lower levels



2. Visual similarity computation

Efficient Computation for High Resolution Images:

All pairs correlation:

- $O(N^2)$, where N is the number of pixels,
- only needs to be computed once and is constant in the number of iterations M

an equivalent implementation of our approach which scales $O(NM)$

at level m, $\mathbf{C}^m_{ijkl}$, and feature maps $g^{(1)} = g_\theta(I_1)$, $g^{(2)} = g_\theta(I_2)$:

$$\mathbf{C}^m_{ijkl} = \frac{1}{2^{2m}} \sum_p^{2^m} \sum_q^{2^m} \langle g^{(1)}_{i,j}, g^{(2)}_{2^m k+p, 2^m l+q} \rangle = \langle g^{(1)}_{i,j}, \frac{1}{2^{2m}} (\sum_p^{2^m} \sum_q^{2^m} g^{(2)}_{2^m k+p, 2^m l+q}) \rangle$$

- we do not precompute the correlations,
  but instead precompute the pooled image feature maps

3. Iterative updates

Our update operator estimates a sequence of flow estimates $\{\mathbf{f}_1, ..., \mathbf{f}_N\}$ from an initial starting point $\mathbf{f}_0 = \mathbf{0}$. With each iteration, it produces an update direction $\Delta\mathbf{f}$ which is applied to the current estimate: $\mathbf{f}_{k+1} = \Delta\mathbf{f} + \mathbf{f}_{k+1}$.

$$\mathcal{L} = \sum_{i=1}^{N} \gamma^{i-N} ||\mathbf{f}_{gt} - \mathbf{f}_i||_1$$

Update: A core component of the update operator is a gated activation unit based on the GRU cell, with fully connected layers replaced with convolutions:

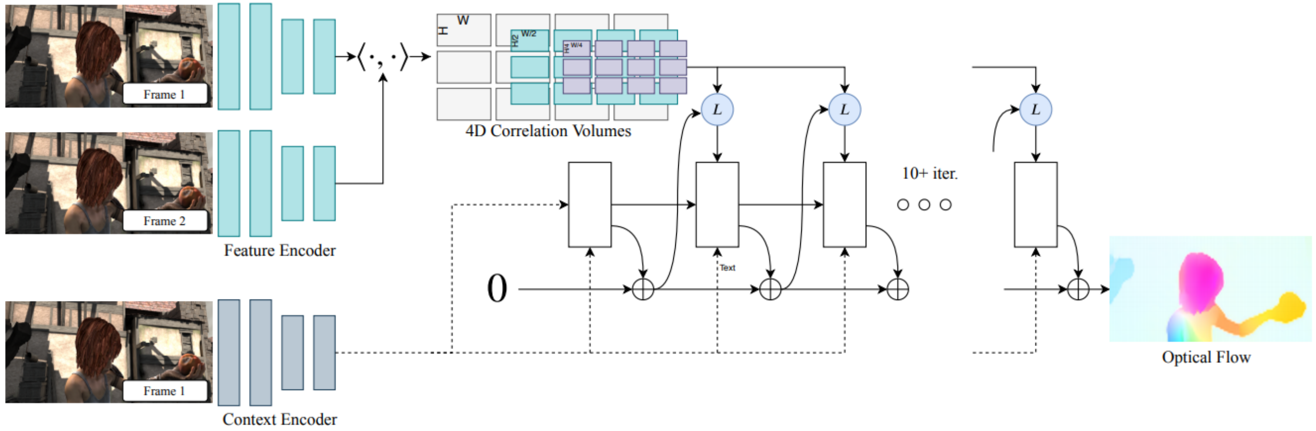
$$z_t = \sigma(\mathrm{Conv}_{3\mathrm{x}3}([h_{t-1}, x_t], W_z)) \quad (3)$$

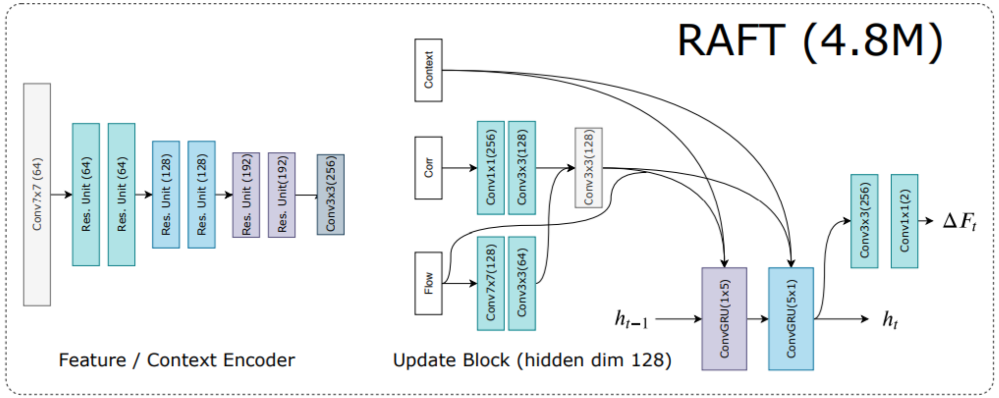
$$r_t = \sigma(\mathrm{Conv}_{3\mathrm{x}3}([h_{t-1}, x_t], W_r)) \quad (4)$$

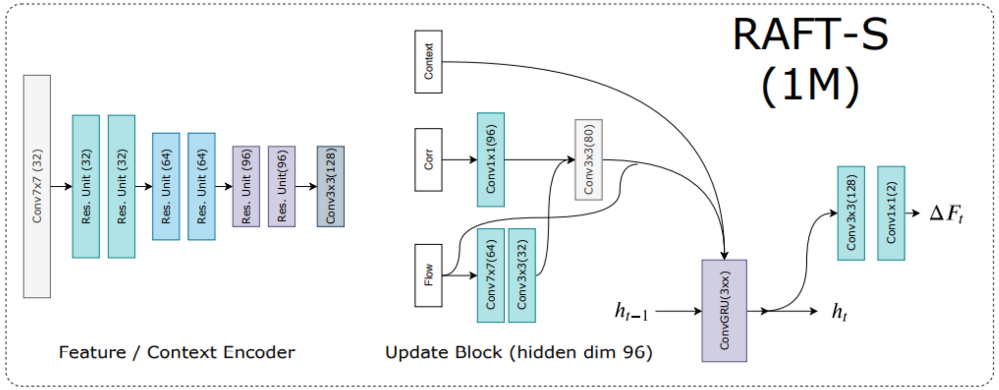
$$\tilde{h}_t = \tanh(\mathrm{Conv}_{3\mathrm{x}3}([r_t \odot h_{t-1}, x_t], W_h)) \quad (5)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (6)$$

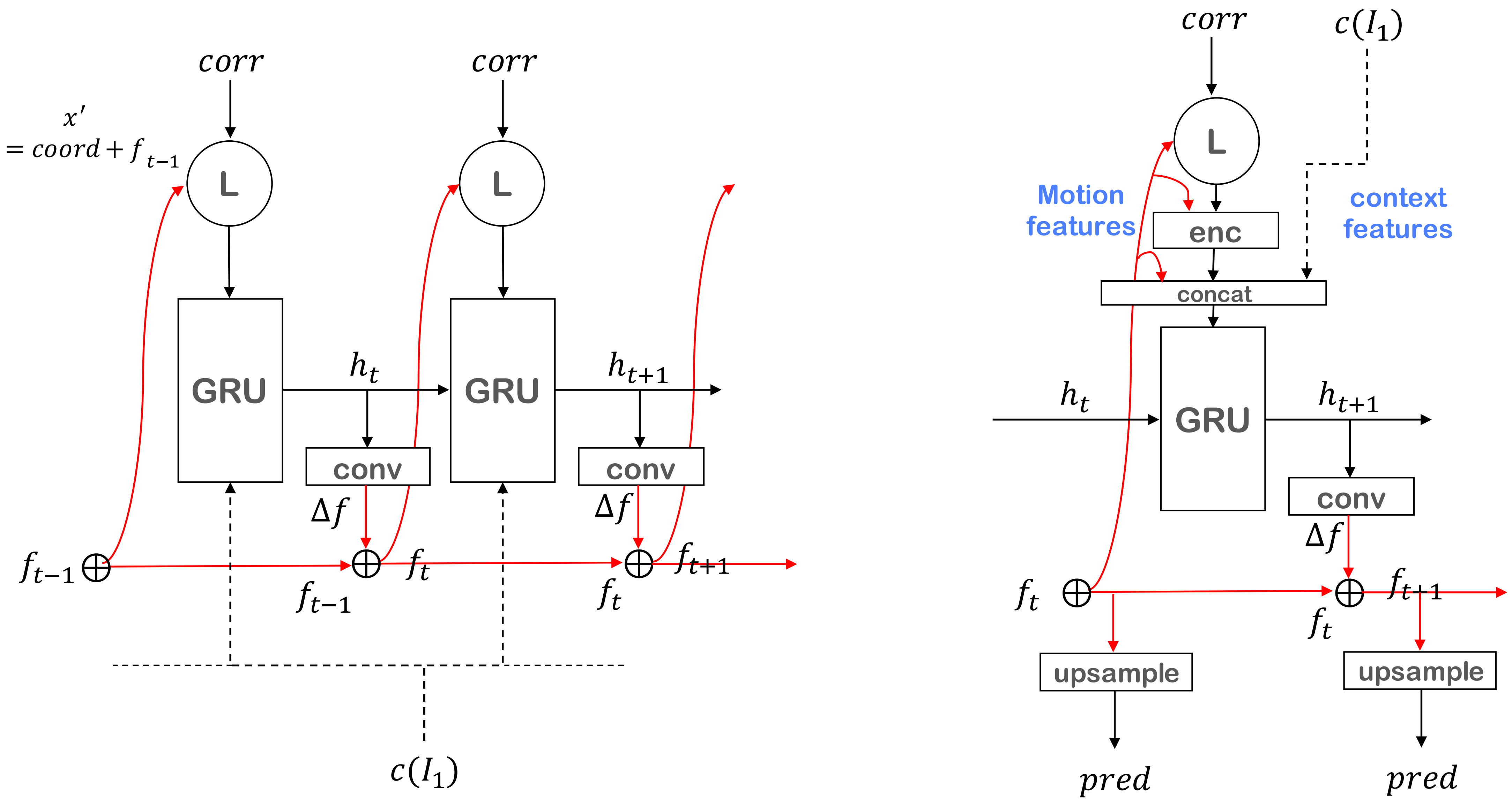
where x_t is the concatenation of flow, correlation, and context features previously







2 Overview


corr
corr
x'
= coord + f t−1
L
L
GRU
GRU
h t
h t+1
conv
conv
Δf
Δf
f t−1
f t−1
f t
f t
f t+1
c(I1)
corr
c(I1)
L
Motion features
context features
enc
concat
h t
GRU
h t+1
conv
Δf
f t
f t
f t+1
upsample
upsample
pred
pred

3. Iterative updates

Upsampling Module

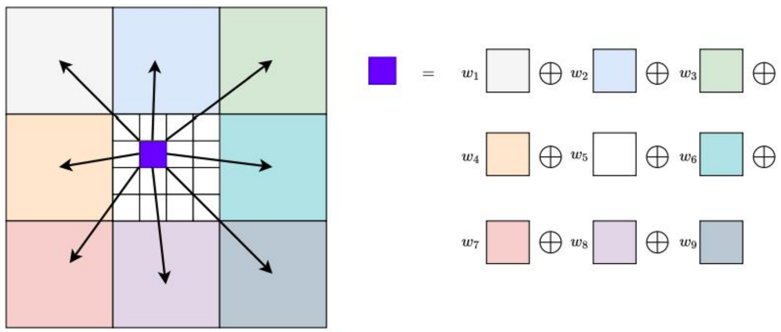


Fig. 2: Illistration of the upsampling module. Each pixel of the high resolution flow field (small boxes) is taken to be the convex combination of its 9 coarse resolution neighbors using weights predicted by the network.

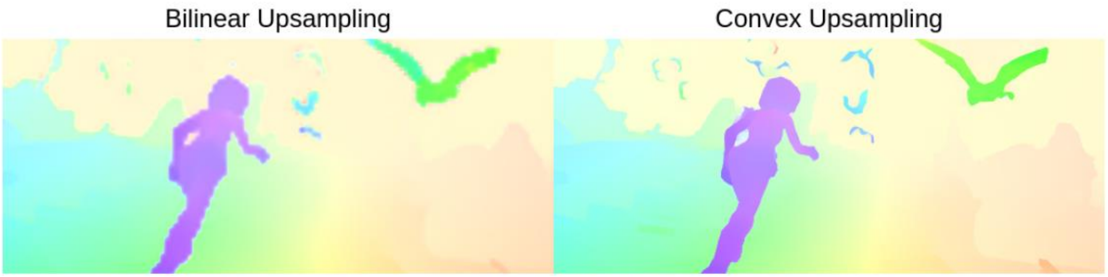


Fig. 3: Our upsampling module improves accuracy near motion boundaries, and also allows RAFT to recover the flow of small fast moving objects such as the birds shown in the figure.

3 Experiments

Fig. 3: Flow predictions on the Sintel test set.

Fig. 4: Flow predictions on the KITTI test set.

Stage	Weights	Training Data	Learning Rate	Batch Size (per GPU)	Weight Decay	Crop Size
Chairs	-	C	4e-4	6	1e-4	[368, 496]
Things	Chairs	T	1.2e-4	3	1e-4	[400, 720]
Sintel	Things	S+T+K+H	1.2e-4	3	1e-5	[368, 768]
KITTI	Sintel	K	1e-4	3	1e-5	[288, 960]

Table 1: Details of the training schedule. Dataset abbreviations: C: FlyingChairs, T: FlyingThings, S: Sintel, K: KITTI-2015, H: HD1K. During the sintel Fine-tuning phase, the dataset distribution is S(.67), T(.12), K(.13), H(.08).

3 Experiments

Training Data	Method	Sintel (train)		KITTI-15 (train)		Sintel (test)		KITTI-15 (test)
		Clean	Final	F1-epe	F1-all	Clean	Final	F1-all
-	FlowFields[7]	-	-	-	-	3.75	5.81	15.31
-	FlowFields++[40]	-	-	-	-	2.94	5.49	14.82
S	DCFlow[47]	-	-	-	-	3.54	5.12	14.86
S	MRFlow[46]	-	-	-	-	2.53	5.38	12.19
C + T	HD3[49]	3.84	8.77	13.17	24.0	-	-	-
	LiteFlowNet[22]	2.48	4.04	10.39	28.5	-	-	-
	PWC-Net[42]	2.55	3.93	10.35	33.7	-	-	-
	LiteFlowNet2[23]	2.24	3.78	8.97	25.9	-	-	-
	VCN[48]	2.21	3.68	8.36	25.1	-	-	-
	MaskFlowNet[51]	2.25	3.61	-	<u>23.1</u>	-	-	-
	FlowNet2[25]	<u>2.02</u>	3.54[1]	10.08	30.0	3.96	6.02	-
	Ours (small)	2.21	<u>3.35</u>	<u>7.51</u>	26.9	-	-	-
	Ours (2-view)	**1.43**	**2.71**	**5.04**	**17.4**	-	-	-
C+T+S/K	FlowNet2 [25]	(1.45)	(2.01)	(2.30)	(6.8)	4.16	5.74	11.48
	HD3 [49]	(1.87)	(1.17)	(1.31)	(4.1)	4.79	4.67	6.55
	IRR-PWC [24]	(1.92)	(2.51)	(1.63)	(5.3)	3.84	4.58	7.65
	VCN [48]	(1.66)	(2.24)	(1.16)	(4.1)	2.81	4.40	**6.30**
	ScopeFlow[8]	-	-	-	-	3.59	4.10	6.82
	Ours (2-view, bilinear)	(1.09)	(1.53)	(1.07)	(3.9)	<u>2.77</u>	<u>3.61</u>	**6.30**
	Ours (warm-start, bilinear)	(1.10)	(1.61)	-	-	**2.42**	**3.39**	-
C+T+S+K+H	LiteFlowNet2[2] [23]	(1.30)	(1.62)	(1.47)	(4.8)	3.45	4.90	7.74
	PWC-Net+[41]	(1.71)	(2.34)	(1.50)	(5.3)	3.45	4.60	7.72
	MaskFlowNet[51]	-	-	-	-	2.52	4.17	<u>6.10</u>
	Ours (2-view)	(0.76)	(1.22)	(0.63)	(1.5)	<u>1.94</u>	<u>3.18</u>	**5.10**
	Ours (warm-start)	(0.77)	(1.27)	-	-	**1.61**	**2.86**	-

Experiments

Experiment	Method	Sintel (train) Clean	Sintel (train) Final	KITTI-15 (train) F1-epe	KITTI-15 (train) F1-all	Parameters
Reference Model (bilinear upsampling), Training: 100k(C) → 60k(T)						
Update Op.	ConvGRU	1.63	2.83	5.54	19.8	4.8M
	Conv	2.04	3.21	7.66	26.1	4.1M
Tying	Tied Weights	1.63	2.83	5.54	19.8	4.8M
	Untied Weights	1.96	3.20	7.64	24.1	32.5M
Context	Context	1.63	2.83	5.54	19.8	4.8M
	No Context	1.93	3.06	6.25	23.1	3.3M
Feature Scale	Single-Scale	1.63	2.83	5.54	19.8	4.8M
	Multi-Scale	2.08	3.12	6.91	23.2	6.6M
Lookup Radius	0	3.41	4.53	23.6	44.8	4.7M
	1	1.80	2.99	6.27	21.5	4.7M
	2	1.78	2.82	5.84	21.1	4.8M
	4	1.63	2.83	5.54	19.8	4.8M
Correlation Pooling	No	1.95	3.02	6.07	23.2	4.7M
	Yes	1.63	2.83	5.54	19.8	4.8M
Correlation Range	32px	2.91	4.48	10.4	28.8	4.8M
	64px	2.06	3.16	6.24	20.9	4.8M
	128px	1.64	2.81	6.00	19.9	4.8M
	All-Pairs	1.63	2.83	5.54	19.8	4.8M
Features for Refinement	Correlation	1.63	2.83	5.54	19.8	4.8M
	Warping	2.27	3.73	11.83	32.1	2.8M
Reference Model (convex upsampling), Training: 100k(C) → 100k(T)						
Upsampling	Convex	1.43	2.71	5.04	17.4	5.3M
	Bilinear	1.60	2.79	5.17	19.2	4.8M
Inference Updates	1	4.04	5.45	15.30	44.5	5.3M
	3	2.14	3.52	8.98	29.9	5.3M
	8	1.61	2.88	5.99	19.6	5.3M
	32	1.43	2.71	5.00	17.4	5.3M
	100	1.41	2.72	4.95	17.4	5.3M
	200	1.40	2.73	4.94	17.4	5.3M

Table 2: Ablation experiments. Settings used in our final model are underlined. See Sec. 4.3 for details.

3 Experiments

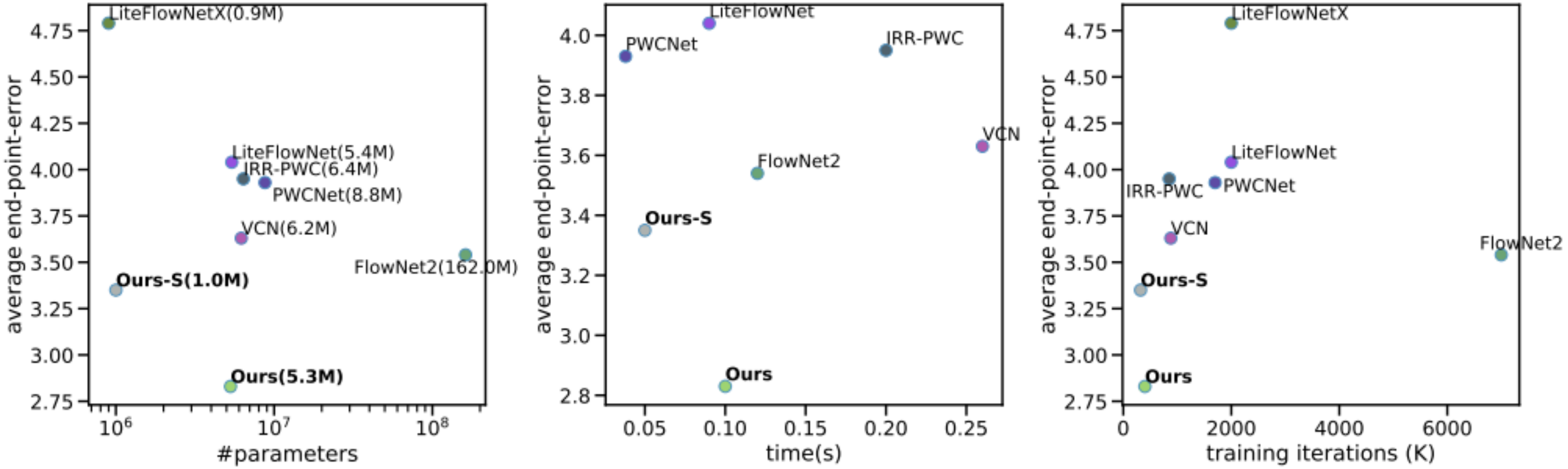


Fig. 6: Results on 1080p (1088x1920) video from DAVIS (550 ms per frame).

Q&A