FDFlowNet: Fast Optical Flow Estimation using a Deep Lightweight Network

Lingtong Kong, Jie Yang
Institute of Image Processing and Pattern Recognition
Shanghai Jiao Tong University

International Conference on Image Processing (ICIP) 2020

1.Introduction

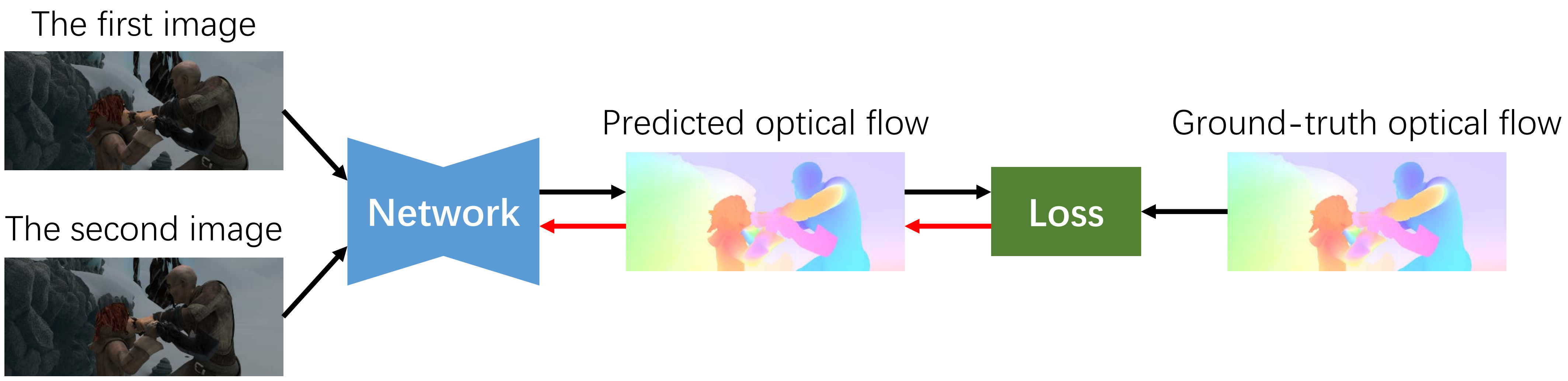
- **Optical Flow Estimation**
  - Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene.
  - A fundamental problem in computer vision which plays an important role in many vision applications such as action recognition, video understanding and self-driving cars.

- **Traditional Methods**
  - Variational approaches based on optimizing an energy function with brightness consistency assumption.
  - Weakness:
    - (1) Requires iteration and is computationally heavy.
    - (2) Dissatisfy real-time applications.
    - (3) Not accuracy in large displacement and occluded areas.

1.Introduction

- **Deep Learning Methods**
  - Given time adjacent image pair and optical flow ground-truth label, training specific deep networks in an end-to-end manner.
  - Advantages:
    - (1) High accuracy in both non occluded and occluded regions.
    - (2) Fast inference speed aided by advanced parallel computing devices, such as GPU.

Flow diagram of training optical flow network



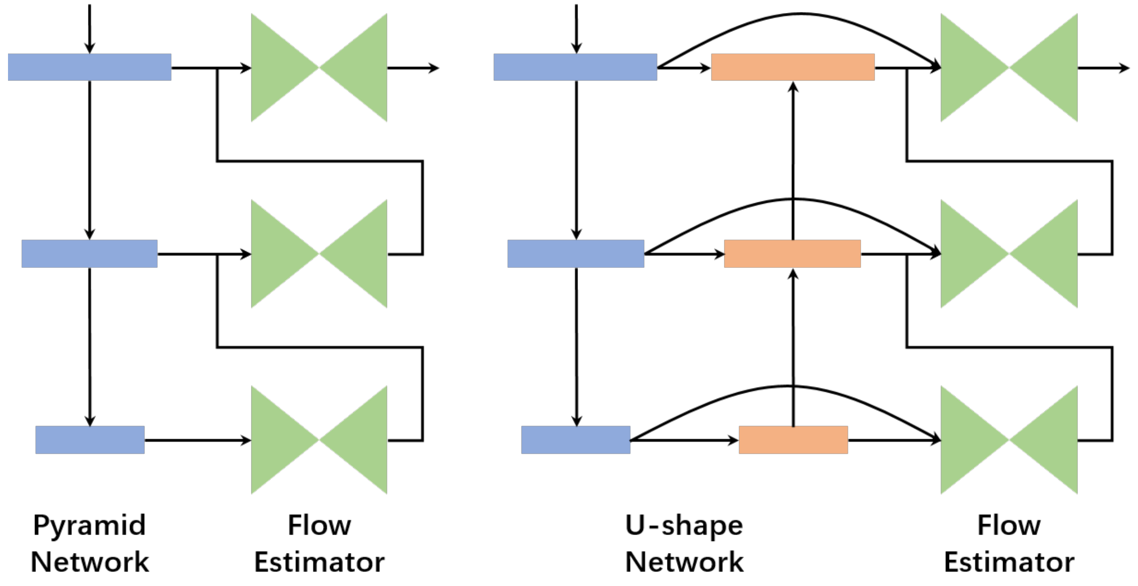
2.Methods

- **Challenges**
  - As part of many vision systems, existing optical flow networks usually take considerable computation that slows down running speed.
  - Many optical flow networks have large number of parameters which is not applicable to mobile applications.
- **Contributions**
  - FDFlowNet: A fast and lightweight deep neural network for efficient optical flow estimation.
    - (1) A compact **U-shape network** for extracting multi-scale context feature compared with existing pyramid network.
    - (2) A **partial fully connected** structure for optical flow decoder that gets a good balance between accuracy and speed.
    - (3) **Cost volume aggregation** and dilated convolution improvement.

Please refer to our paper for network details.

2.Methods

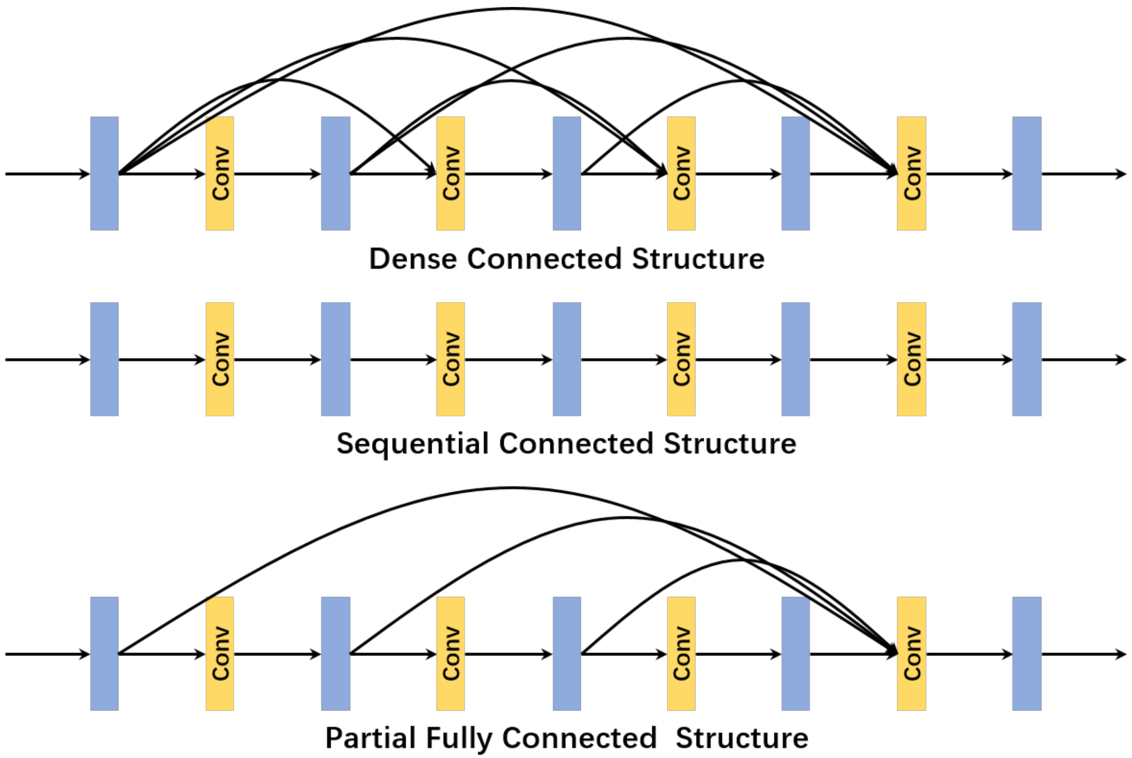
- **U-shape Network**
  - The left is traditional pyramid network and the right is our U-shape network.
  - A drawback of pyramid network is that the higher resolution where flow estimator locates, the less semantic information corresponding pyramid feature contains. So it takes more convolution layers for the flow estimator to rebuild it. On the other hand, this semantic information is similar among different resolutions, which means that computation on multiple levels can be redundant.



- The proposed U-shape backbone can fuse better multi-scale information.
- The fused feature is only replacing context feature. And pyramid feature is retained for building cost volume to keep local properties.

2.Methods

- **Partial Fully Connected Flow Estimator**
  - Comparison of three different modes of flow estimator:
    - Sequential Connected
    - Dense Connected
    - **Partial Fully Connected**
  - Proposed partial fully connected structure provides a balance and tradeoff between the other two types in model size, computation cost and network performance.



- The last two sequential convolution layers are changed to dilated convolution with dilation rate 2 for enlarging receptive field.

3.Experiments

- **Training Strategy**

  All experiments are conducted on one NVIDIA 1080Ti GPU. We implement codes in PyTorch and adopt Adam optimizer. Weight decay is set to 1e-4 for regularization.

  - FlyingChairs:
    - Batch size 8, Crop size 320 x 480
    - Learning rate 1-300k: 1e-4, 301-400k: 5e-5, 401-500k: 2.5e-5, 501-600k: 1.25e-5
  - FlyingThings3D:
    - Batch size 4, Crop size 384 x 768
    - Learning rate 1-200k: 1e-5, 201-300k: 5e-6, 301-400k: 2.5e-6, 401-500k: 1.25e-6
  - MPI Sintel:
    - Batch size 4, 2 from Clean and 2 from Final, Crop size 384 x 768
    - Learning schedule is the same as PWC-Net.
  - KITTI:
    - Batch size 4, 2 from kitti 2012 and 2 from kitti 2015, Crop size 320 x 896
    - Learning schedule is the same as PWC-Net while reducing data augmentation.

3.Experiments

- **Benchmark Results**

Method	Sintel Clean		Sintel Final		KITTI12			KITTI15		
	train	test	train	test	train	test	test(Fl-Noc)	train	train(Fl-all)	test(Fl-all)
FlowNetC [6]	4.31	7.28	5.87	8.81	9.35	-	-	-	-	-
FlowNetC-ft	(3.78)	6.85	(5.28)	8.51	8.79	-	-	-	-	-
SPyNet [8]	4.12	6.69	5.57	8.43	9.12	-	-	-	-	-
SPyNet-ft	(3.17)	6.64	(4.32)	8.36	(4.13)	4.7	12.31%	-	-	35.07%
FlowNet2 [7]	2.02	3.96	3.14	6.02	4.09	-	-	10.06	30.37%	-
FlowNet2-ft	(1.45)	4.16	(2.01)	5.74	(1.28)	1.8	4.82%	(2.30)	(8.61%)	11.48%
PWC-Net [9]	2.55	-	3.93	-	4.14	-	-	10.35	33.67%	-
PWC-Net-ft	(2.02)	4.39	(2.08)	**5.04**	(1.45)	1.7	4.22%	(2.16)	(9.80%)	9.60%
PWC-Net-small [9]	2.83	-	4.08	-	-	-	-	-	-	-
PWC-Net-small-ft	(2.27)	5.05	(2.45)	5.32	-	-	-	-	-	-
LiteFlowNet[10]	2.48	-	4.04	-	4.00	-	-	10.39	28.50%	-
LiteFlowNet-ft	**(1.35)**	4.54	**(1.78)**	5.38	**(1.05)**	1.6	3.27%	(1.62)	**(5.58%)**	**9.38%**
FDFlowNet	2.60	-	4.12	-	4.13	-	-	10.75	29.59%	-
FDFlowNet-ft	(1.80)	**3.71**	(1.93)	5.11	(1.09)	**1.5**	**3.19%**	**(1.56)**	(6.36%)	**9.38%**

- Our FDFlowNet ranks the first among listed deep networks on Sintel Clean test, KITTI 2012 test and KITTI 2015 test datasets.

3.Experiments

- **Ablation Study**
  - "FDFlowNet-U" means that U-shape network is removed. "FDFlowNet-PFC" represents substituting sequential connected structure for partial fully connected structure with dilated convolution.
  - All models are trained on FlyingChairs using the same learning schedule and evaluated on Chairs test, Sintel training datasets.
  - Experiments show that U-shape network can provide better feature representation with fused multi-scale information that obtains an obvious improvement. It is about **5.6%** improvement on Sintel Clean and about **5.2%** improvement on Sintel Final. Partial fully connected structure with dilated convolution also surpass traditional sequential topology.

Variants	FDFlowNet	FDFlowNet-U	FDFlowNet-PFC
Chairs	**1.92**	2.14	2.02
Sintel Clean	**3.06**	3.24	3.18
Sintel Final	**4.23**	4.46	4.34

3.Experiments

- **Runtime and Parameters**
  - It is important for optical flow network to be running fast in real-time and lightweight. This is especially significant in embedded and mobile devices.
  - Experiments are conducted on a machine equipped with one NVIDIA GTX 1080Ti GPU. We use the PyTorch implement of all networks for fair comparison. Running time is obtained on Sintel resolution 436 x 1024 averaged over 1000 times.
  - Our FDFlowNet **runs fastest** among all the well-behaved models. It is about **2 times faster** than PWC-Net and about **3.2 times faster** than LiteFlowNet. It also outperforms PWC-Net-small in both running speed and benchmark performance.

Model	FlowNetC	FlowNet2	SPyNet
parameters (M)	39.18	162.49	**1.20**
runtime (ms)	24.6	115.7	47.4
Model	PWC-Net	LiteFlowNet	FDFlowNet
parameters (M)	8.75	5.37	5.79
runtime (ms)	32.2	53.2	**16.7**

3.Experiments

- Video Demos of proposed FDFlowNet

Sintel

KITTI

Thanks for watching the presentation of our ICIP 2020 paper, FDFlowNet: Fast Optical Flow Estimation using a Deep Lightweight Network.