Dense Optical Flow using RAFT

Mohd Asif Khan
Information Technology
National Institute of Technology, Karnataka
Surathkal, Mangalore, Karnataka, India
asifk.202it013@nitk.edu.in

Praful kumar
Information Technology
National Institute of Technology, Karnataka
Surathkal, Mangalore, Karnataka, India
praful.202it020@nitk.edu.in

Mr Dinesh Naik
Information Technology
National Institute of Technology, Karnataka
Surathkal, Mangalore, Karnataka, India
din_nk@nitk.edu.in

***Abstract*—In this project we have used an approach called Recurrent All-Pairs Transform, also known as RAFT. This is a deep network architecture for the detection of optical flow in the images. The RAFT model relates the per pixel motion between images even for minor changes in the position of the objects. It also updates the flow of field through recurrent units that perform lookups on the performance of the model. RAFT also works well with different datatypes and also it has better efficiency, training speed and count of parameters. We have performed the experimentation by using different parameters and also by changing certain values in the model itself. We have used one cycle learning to find the best parameters for the model. We also found that the RAFT model performs better than most of the other existing models for optical flow calculation in to images.**

***Index Terms*—Optical flow, correlation volume, flow field**

I. Introduction

Optical flow is the problem of determining the per-pixel motion between two frames of videos or images. This has gained attention of many data scientists because of certain challenges that are faced in solving the problem. Certain challenges that are faced in finding optical flow are fast-moving objects, occlusions, blur due to motion, textureless images. Due to the problem faced in solving the optical flow the problem is really complex and also interesting.

Traditionally the optical flow problem has been approached as a optimization problem over the dense displacement fields between two images. Also, there is a trade-off between data and regularization as data gives us information regarding how similar the images are and the regularization gives us the information regarding the possibility of motion between two frames. By using this approach good results were obtained, but improving those results further very challenging because this approach was hand-designed and also robust on many of the corner cases.

Certain deep learning models have been developed recenetly that show a comparable performance to the traditional method as discussed above. These models can replace the traditional method and can also directly predict the optical flow. These models also have an advantage that they can achieve the better inference time as compared to the traditional method and the performance is comparable to the traditional method.

Here, we have used a deep learning architecture called Recurrent All-Pairs Field Transforms(RAFT) [1] in order to predict the optical flow. This model has many advantages even when compared to the other models. Accordibng to the base paper followed, RAFT [1] achieves 16% better F1-all error and 30% better end-point-error from the previous best published results. Also, when trained on the synthetic data, it achieves 40% error reduction from the previous best works. RAFT trains 10 times faster than most of the other deep learning models. The dataset used in these experiments are *KITTI* and *Sintel.*

The RAFT [1] model consists of three main components: feature encoder, correlation layer and a GRU-based update operator. Feature encoder is responsible for extracting vectors for each pixel, the correlation layer produces 4D correlation volume for all the pairs of pixel and the GRU-based update operator retrieves values from the correlation volumes and updates a flow field which was initialized to zero.

The architecture of RAFT [1] is inspired from the traditional optimization-based approaches as here the feature encoder extracts per-pixel features, correlation layer computes the visual similarity between the pixels and the update operator works like in iteration. Like the traditional approach, features and motion priors are not hand-crafted here, but instead they are leaned by the encoder and update operator respectively.

The design of RAFT [1] is similar to many of the existing models, but it has it's own unique features, so the design can also be considered as novel. RAFT [1] updates the single flow field in a high resolution directly unlike the previous work where flow is first estimated at a lower resolution and then later it is upscaled. Due to computation in high resolution RAFT [1] can accurately measure the optical flow even in the fast moving objects. Also, the update operator in the RAFT is also lightweight because in the other proposed models they do not tie weights across iterations and are also limited in the number of iterations. The other proposed models can be applied for upto 5 iterations and also are limited by size

of network, RAFT [1] can more than 100 iterations and is not very much limited by the size of the network. Even the update operator of RAFT has a unique design that consists of convolutional GRU and performs and performs 4D multiscale correlation volumes unlike the previous models that had simple convolutional networks.
The experiments performed for the optical flow are on the *Sintel* and *KITTI* datasets. We have performed experiments o=on the different parameter and have also used one cycle learning policy whoch says that cyclically varying the learning rates between the reasonable bounds can increase the accuracy of the model.

II. Related work

A lot of work has been done in the field of determining the Optical flow. A few of them that we have studied are discussed in this section further.
The PWC-net [2] uses several classic optical flow techniques like image warping, cost-colume and end-to-end trainable deep learning model. These are the simple and well defined principles for the optical flow problem. By using the learnable feature pyramid, PWC-Net uses the optical flow estimate to comapare features with the second image. It uses warped features and features from first image to calculate the optical flow. This model is 17 times smaller in size as compared to the FlowNet2 model which was state of the art at the time of its development, this smaller size makes it faster to train. It has even outperformed on the KITTI2015 dataset. Although the FlowNet2 achives an impressive performance by stacking several models into a large-capacity model this PWC-Net obtains the comparable performance by embedding the classical performance into the network architecture. Also, it would be interesting to use the PWC-Net as a building block to design a large architecture.
The FlowNet2.0 [3] shows that the schedule of presenting data during training is very important. It also developed a stacked architecture that includes warping of second image with intermediate optical flow. They also introduced specializing on small networks for the small movements. This showed a better performance when compared to the FlowNet model and also provided a network running from 8 to 140fps. This performed good results on the Sintel dataset, but the results on the KITTI dataset were not satisfactory.
Improving optical flow on a pyramidal level [4] presented the concept of spatial feature pyramid in a modern, deep learning based algorithms. This departed from a warping-to a sampling based strategy to overcome issues like handling large motions from small objects and even eliminated noise that was for improved convergence and a better performance. Although this model produced brilliant results, they were second best for the KITTI2015 and Sintel datasets.
Liteflownet [5] is a lightweight optical flow model for the prediction of optical flow. It proposes that more effective flow at each pyramid level through a lightweight cascaded network which not only flow estimation accuracy, but also permits seamless incorporation of descriptor matching in the network. Also, the effective structure by the pyramidal model embraces feature warping rather than image warpinf as in [3]. Also, it was 30 times smaller and 1.3 times faster when compared to the FlowNet2.0 model but it couldn't achieve accuracy comparable to the FlowNet2.0 model.
The DeepFlow model [6] is a descriptor matching algorithm which uses dense sampling for the retrieval of the dense correspondences from single feature corresponding to the deformable patches. Also, the matching algorithm works with the restricted set of feasible non-rigid warpings which gracefully produces almost smooth dense correspondences while allowing computationally efficient comparison of non-rigid descriptors. This model handles large displacements very well and shows a competitive performance and shows state-of-the art results for the Sintel dataset.
Volumetric correspondence networks for optical flow [7] shows 4D for convolutional matching module for volumetric correspondence processing also factorizing the filter into separable components that are implemented with an encoder-decoder and also one can significantly reduce computation and memory. Also, it integrate volumetric filtering into a coarse-to-fine warping scheme, where ambiguous matches and coarse-mistakes are handled by the multi-hypotheses design. Compared to the previous state of the art approach, this was more accurate, easier to train and also generalizes better. This was possible because of volumetric encoder-decoder layers, multi-channel cost volumes, and separable volumetric filters. But, due to the limitations of the CUDA kernel and hardware support for convolutions and poolings with non-standard shapes, the FLOPS numbers for the current implementation are not directly transferable to running time.
Hierarchical discrete distribution decomposition for match density estimation [8] has shown a framework suitable for learning probabilistic pixel correspondences in both optical flow and stereo matching. Full match density is decomposed to multiple scales hierarchically and the local matching distributions at each scale are estimated and conditioned on the matching and warping at coarser scales. This model did not produce any information on the assignment probabilities from the segmentation and also no relationship between the match densities of the adjacent pixels.
The FlowNet model [9] proposed a model by using the Convolution neral network and approaching the optical flow problem as a supervised learning task. Two architectures were compared: a generic one and one including a layer that correlates feature vectors at different image locations. Also, the synthetic FlyingChairs dataset was generated and used as ground truth dataset was not sufficient for the training of large CNN. Also, the conclusion was made that networks trained on this unrealistic data still generalize very well to existing datasets such as Sintel and KITTI, achieving competitive accuracy at frame rates of 5 to 10 fps.
Reliable Supervision from Transformations for Unsupervised Optical Flow Estimation [10] discusses the supervised learning for optical flow, which leverages the supervision from the view of synthesis and the objective of unsupervised learning

is likely to be unreliable in challenging scenes. This presents a framework to use more reliable supervision from the tranformations. It twists the general unsupervised learning pipeline by running another forward pass with transformed data from augmentation, along with using transformed predictions of original data as the self-supervision signal. Also, it introduces a lightweight network with multiple frames by a highly-shared flow decoder. The method got best accuracy among the results present at that time.

MaskFlownet [11] proposes the model to solve the ambiguity caused by occluded area during image warping. This propose an asymmetric occlusion-aware feature matching module, which can learn a rough occlusion mask that filters useless (occluded) areas immediately after feature warping without any explicit supervision. The proposed module can be easily integrated into end-to-end network architectures and enjoys performance gains while introducing negligible computational cost. The learned occlusion mask can be further fed into a subsequent network cascade with dual feature pyramids with which state-of-the-art performance was achieved.

III. Methodology

The aim here is to calculate the dense displacement field of a given pair of images by mapping the pixels of the pixels in second image to the corresponding coordinates in the second image itself. The methodology is divided into 3 basic steps:

- feature extraction
- computing visual similarity
- iterative updates

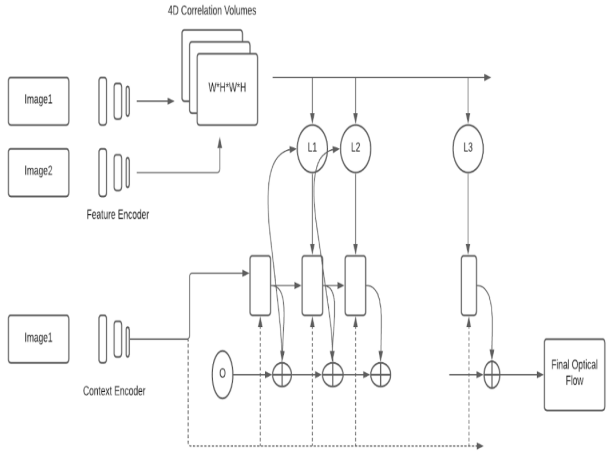


Fig. 1: RAFT methodology

Let us consider the images as I1 and I2 as the two input images. For this given pair of images, we need to calculate the dense displacement field which maps each pixel (u,v) in I2 to it's corresponding coordinates (u',v') = (u+f^1(u), v+f^2(v)) int I2. Now, each of the three steps is taken individually and explained further.

A. Feature extraction

The feature extraction is done using the Convolutional network. The feature encoder network is applied to both the input images and mapping between the input image and dense feature maps is done at the lower resolution. The encoder that we are applying outputs the features at 1/8 resolution. The feature encoder consists of 6 residual blocks 2 at 1/2 resolution, 2 at 1/4 resolution and 2 at 1/8 resolution. We also have a context network that extracts features only from the first image. The architecture for this is similar to the architecture of the feature extraction network. This Feature extraction stage is only performed once in the whole process.

B. Computing visual similarity

Visual similarity is calculated by the construction of the full correlation between all the pairs. The correlation volume is is constructed by calculating the dot product between the pairs of feature vectors. The correlation volume can be effectively computed as a single matrix multiplication. The concepts used here are explained below:

1) Correlation pyramid: A 4-layer pyramid is constructed by pooling the last two dimensions of the correlation volume with kernel sizes 1, 2, 4 and 8 and an equivalent stride. These set of volumes gives information regarding the large and small displacements. By maintaining the first 2 dimensions we maintain high resolution information, allowing the model to recover the motions of fast moving objects.

2) Correlation Lookup: The lookup operator generates the feature map by indexing from the correlation pyramid. For a current estimate of optical flow, we map each pixel in I1 to its estimated correspondence in I2. A local grid is defined around this estimated correspondence as the integer offsets which are within a radius r from it using the L1 distance. We use local neighbour to index the correlation volume. Lookups on all levels of pyramid are formed. The values from all the levels are concatenated into a single feature map.

C. Iterative updates

The update operator estimates a sequence of flow from the initial starting point. After every iteration, there is an update direction produced which is applied to the current estimate. This update operator takes the flow, correlation and a latent hidden state as input and outputs the update along with the updated hidden state. The steps of the optimization algorithm are bring copied here. Also, we have used the tied weights across depth and also the convergence to a fixed point. Finally the update operator is trained to perform the updates such that sequence converges to a fixed point. The concepts used here are explained below:

1) Initialization: The initialization of the flow field is 0 by default, but our model gives us freedom to initaialize according to our choice.

2) Inputs: Given the current flow estimate, correlation features are retrieved from the correlation pyramid. The correlation features are then processed by 2 convolutional layers. Additionally, 2 convolutional layers are applied to the flow estimate itself to generate flow features. Finally, directly inject the input from the context network. The input feature map is then taken as the concatenation of the correlation, flow, and context features.

3) Update: this is the most important component and it is a gated activation unit based on the GRU cell, with fully connected layers replaced with convolutions.

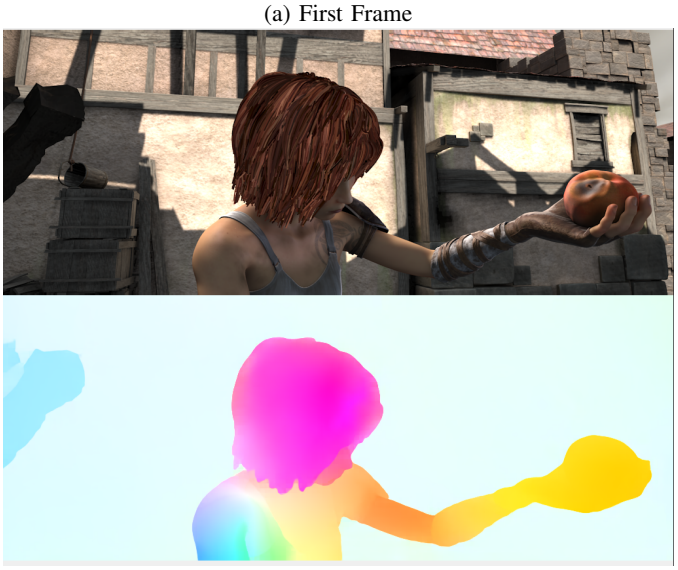
4) Flow prediction: The hidden state outputted by the GRU is passed through two convolutional layers to predict the flow update. The output flow is at 1/8 resolution of the input image. During training and evaluation upsampling of the predicted flow is done in order to match the ground truth.

5) Unsampling: The network updates the optical flow at 1/8resolution. Upsampling of the optical flow is done to full resolution by taking the full resolution flow at each pixel to be the convex combination of a 3x3 grid of its coarse resolution neighbors.

IV. Demo

After running the trained model on images we got the optical flow outputs like shown in figure 2.

(a) First Frame



(b) Second Frame

Fig. 2: Dense optical flow on 2 consecutuve frames

V. Experiments

Even though RAFT [1] is state-of-the-art model but still we found out that it was taking too much time on Google Colab. We tried to take the small version of RAFT which is having 990162 parameters and tried to decrease the end point error as well as reducing the time taken to train the model. We instead of doing all 100,000 steps did only 10,000 steps on sintel dataset. Our goal was getting best accuracy on sintel dataset while modifying the architecture.

The parameters on which we experiment on are :

- Batch Size
- Learning rate
- Weight decay
- Gamma
- GRU iterations
- Steps
- Cycle momentum
- Radius

Manual Hyperparameter Tuning							
Batch Size	Learning rate	Weight decay	gamma	GRU iterations	steps	Sintel (clean) EPE	Sintel (final) EPE
10	0.000125	0.00001	0.9	12	5000	4.25	5.05
10	0.000125	0.00001	0.9	12	10000	3.50	4.23
5	0.00125	0.00001	0.9	12	5000	3.76	4.52
10	0.00125	0.00001	0.9	12	5000	2.43	3.16
5	0.0001	0.00001	0.9	12	5000	5.67	6.44
5	0.0001	0.00001	0.9	12	5000	3.83	4.54
10	0.000125	0.00001	0.9	12	5000	4.25	5.05
10	0.000125	0.00001	0.9	12	10000	3.38	4.22
13	0.000125	0.00001	0.85	6	5000	5.16	5.19
13	0.000125	0.00001	0.85	6	10000	4.06	4.79

After looking at the authors result from RAFT [1] not much improvement was coming so we moved to other ways to improve the results further. We moved to One-Cycle learning policy as discussed in [12].

A. One-Cycle Learning

We modified the cycle momentum parameter in the best parameters obtained previously and we got results like this :

One Cycle learning	
Property Name	Property value
Batch Size	13
Learning rate	0.000125
Weight decay	0.00001
gamma	0.85
GRU iterations	6
steps	10000
cycle momentum	true
Sintel (clean) EPE	3.02
Sintel (final) EPE	3.80

After applying cycle momentum in One-Cycle learning [12] the results got better than most of the previous results.

B. Transfer Learning

We also looked at the transfer learning paper [13] and since we had pretrained sintel small model. We trained the model with best parameters obtained previously using transfer learning on pretrained sintel small model.The results were like this :

Transfer Learning							
Batch Size	Learning rate	Weight decay	gamma	GRU it-era-tions	steps	Sintel (clean) EPE	Sintel (fi-nal) EPE
13	0.000125	0.00001	0.85	6	5000	4.25	5.05
13	0.000125	0.00001	0.85	6	10000	2.10	3.26
13	0.000125	0.00001	0.85	12	5000	1.81	2.75
13	0.000125	0.00001	0.85	12	10000	1.60	2.34

After taking the sintel-small model which was trained on flying chairs and flying things dataset. It was further trained on sintel dataset and the results were comparable to RAFT [1] original model and better than the RAFT [1] small model.

C. Architecture Changes

For further improvement we want modify the architecture as well to see how the model accuracies behave on modifying architecture. We tried modifying the number of levels in correlation pyramid but the convolutions are connected to that. So it can not be modified unless we modify the convolutions. Also there was radius parameter in correlation block. So we modified the radius parameter as well to look at its implication on the model. And the results look like this:

Correlation lookup radius tuning		
Radius	Sintel (clean) EPE	Sintel (final) EPE
4	1.60	2.34
3	1.58	2.20
5	1.56	2.29

Based on data given above there is no clear correlation between radius and EPE in Sintel dataset. But since high correlation lookup radius will mean that we can look at longer distance while looking for optical flow of pixel will increase time to search so smaller values of correlation lookup values are preferred.

VI. Conclusion

Based on the experiments we did on the Sintel dataset we got comparative performace to RAFT [1] original model and better performance to RAFT [1] small model. Also the best results were obtained when we used transfer learning on a pretrained model. The model is also trained for less steps than the original number of steps mentioned in [1]. For future work we would like to improve the architecture further by modifying the correlation pyramid as well as GRU iterations part because that is an iterative process and take some time but essential for good prediction of optical flow using context information from first image. In future we will try to improve that pipeline as well.

References

[1] Z. Teed and J. Deng, "RAFT: recurrent all-pairs field transforms for optical flow," *CoRR*, vol. abs/2003.12039, 2020. [Online]. Available: https://arxiv.org/abs/2003.12039

[2] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," 2018.

[3] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," 2016.

[4] M. Hofinger, S. R. Bulò, L. Porzi, A. Knapitsch, T. Pock, and P. Kontschieder, "Improving optical flow on a pyramid level," 2020.

[5] T.-W. Hui, X. Tang, and C. C. Loy, "Liteflownet: A lightweight convolutional neural network for optical flow estimation," 2018.

[6] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[7] G. Yang and D. Ramanan, "Volumetric correspondence networks for optical flow," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/bbf94b34eb32268ada57a3be5062fe7d-Paper.pdf

[8] Z. Yin, T. Darrell, and F. Yu, "Hierarchical discrete distribution decomposition for match density estimation," 2019.

[9] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," 2015.

[10] L. Liu, J. Zhang, R. He, Y. Liu, Y. Wang, Y. Tai, D. Luo, C. Wang, J. Li, and F. Huang, "Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation," 2020.

[11] S. Zhao, Y. Sheng, Y. Dong, E. I.-C. Chang, and Y. Xu, "Maskflownet: Asymmetric feature matching with learnable occlusion mask," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. [Online]. Available: http://dx.doi.org/10.1109/CVPR42600.2020.00631

[12] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay," *CoRR*, vol. abs/1803.09820, 2018. [Online]. Available: http://arxiv.org/abs/1803.09820

[13] K. You, Z. Kou, M. Long, and J. Wang, "Co-tuning for transfer learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 17 236–17 246. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/c8067ad1937f728f51288b3eb986afaa-Paper.pdf