Project 4 : Learning the Structure from Motion, An Unsupervised Approach

Abhilash Pravin Mane
Masters of Engineering in Robotics
University of Maryland, College Park
Email: amane@umd.edu

Advait Patole
Masters of Engineering in Robotics
University of Maryland, College Park
Email: apatole@umd.edu

I. Introduction

In the previous project we developed structure from motion using traditional approach.In that project we had computed depth and camera poses using multiple images of a scene taken from different view.From the project we learnt that it's output was sparse and in order to get a dense SFM output we had to use deep learning methods to get such output.There has been a lot of work in the field of structure from motion using deep learning methods one such paper *Unsupervised Learning of Depth and Ego-Motion from Video* deals with the same.In this project we explore the paper and we try to improve the network using various techniques.Few methods that we implemented worked and gave improved results while few others did not work.In this project we explain each techniques and it's effect on the network and the output.

II. SFM Learner

The SFMLearner tries to predict camera motion and scene structure from camera pose and depth data in an unsupervised manner(without Labels data) and trains itself by minimizing losses and tries to predict the output as close to the ground truth data.In this project we have trained the SFM Learner as well as applied new techniques to improve it's accuracy.

The network comprises of 2 parts that is dedicated to train the depth and the camera pose.The depth network comprises of a DispNet architecture that is mainly based on an encoder-decoder design with skip connections and multi-scale side predictions.ReLU activation function is present after every convolutional layers, except for prediction layers where we use

$$1/(\alpha \times sigmoid(x) + \beta)$$

as activation function.The depth network takes a single image frame as input and outputs the depth map while the pose network takes multiple inputs in the form of target views and source views.The target view is concatenated with multiple source views and it outputs the relative camera poses between the target view and each of the source view.After that the output of both the network is used to inverse warp the source views to get the target views.The photometric loss is used while training the CNNs.The following is the architecture of SFMLearner.

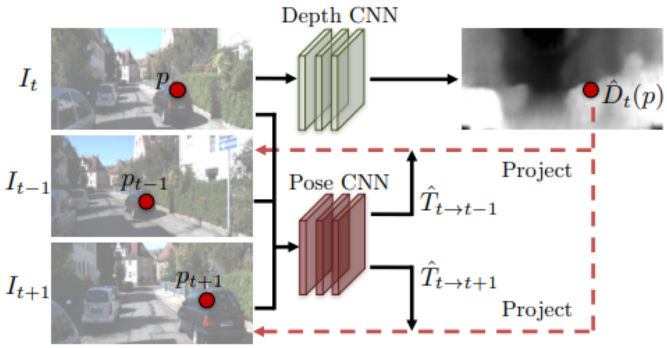
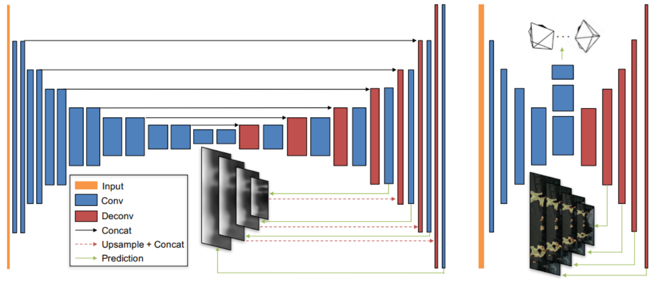


Fig. 1: SFMLearner Architecture



(a) Single-view depth network

(b) Pose/explainability network

Fig. 2: Depth and Pose Network

One of the novel thing about SFMLearner is novel view synthesis ie. given one input view of a scene, to synthesize a new image of the scene seen from a different camera pose.In this way we can predict target view as well as pose and visibility in the nearby using view synthesis which is implemented in a fully differentiable manner with CNNs as pose and depth module.The view synthesis formulation implicitly assumes 1) the scene is static without moving objects 2) there is no occlusion/dis occlusion between the target view and the source views 3) the surface is Lambertian so that the photo-consistency error is meaningful. If any of these assumptions are violated in a training sequence, the gradients could be corrupted and potentially inhibit training. To improve the robustness of our learning pipeline to these factors,there is an additional explain ability prediction network that outputs a per-pixel soft mask E_s for each target-source pair. During training, batch normalization was used for all the layers except for the output layers, and the Adam optimizer with $\beta 1$ = 0.9, $\beta 2$ = 0.999, learning rate of 0.0002 and mini-batch size of 4. The training typically converges after about 150K iterations.The size of images used were 128 x 416 for image sequence.For this project we have used KITTI dataset.The next section deals with the modifications that were done to the SFMLearner to improve the results.

III. Methods used to modify network

A. Structural Similarity Loss

While training the SFMLearner the main objective was to minimize the photometric loss(pixel loss).It measures the high level similarity between target and warped source image.The photometric loss is calculated as the L1 norm between the target image and the warped source images.However this loss makes certain assumptions like scenes should have constant brightness, luminosity.These assumptions does not hold true in every scenario.In order to rectify it we added structural similarity metric(SSIM) to this loss , this performs the same function of L1 norm but it is more efficient because it measures the difference in the images as perceived by humans and is not limited to just pixel level differences.Hence our photometric now becomes the weighted average between the L1 and the SSIM loss which is given by

$$L = \alpha \frac{1 - SSIM(I_t, I_s)}{2} + (1 - \alpha)\|I_t - I_s\|$$

Now while training the network we minimize this loss which is the combination of L1 and SSIM loss.

B. Variable Learning Rate

The learning rate in SFMLearner is 0.0002.In the paper they trained it with a batch size of 4 and it took around 200 iterations before it converges.Due to this it took very long before the values could drop considerably.In our approach we have used a variable learning rate so that the values could drop.We have taken our initial learning rate as 0.002 and we have dropped it by 20% after 60000 iterations (20 epochs)

C. Epipolar Constraints

The problem with simply lowering photometric error is that it ignores ambiguous pixels such those from non-rigid objects, those that are occluded, and so on. As a result, we must weight pixels accordingly depending on whether they are properly projected. One way to assure proper projection is to check if the matching pixel p is on its epipolar line.

$$L_{warp} = \frac{1}{N}\sum_{s}\sum_{p} |I_t(p) - \hat{I}_s(p)| e^{\hat{\tilde{p}}^T E \tilde{p}} \tag{1}$$

The updated photometric loss with the epipolar constraint is shown above. This tweak ensures that the pixel is properly re-projected. If a pixel is projected accurately based on the anticipated depth and posture, the epipolar loss is minimal. However, even if the pixel is suitably distorted, the photometric inaccuracy for a non-rigid object would be significant due to the posture and depth. Incorrect warping might result in significant photometric error. We weight adequately warped pixels with their epipolar distance to avoid punishing them for belonging to moving objects, giving their photometric loss a lower weight than erroneously warped pixels. If the epipolar loss is considerable, the projection is erroneous, resulting in a high weight for the photometric loss.

D. Data Augmentation

Data Augmentation is an important technique that can improve the results of network.Due to availability of data in different format can help the network to train better.SFMLearner itself has some data augmentation present in it like normalization, random scaling , flip and crop.We have added some other data augmentation methods like improving the contrast and brightness of image.The brightness factor is same over all channels of the image and we have selected the factor at random in the range (0.5,2) and we have also implemented gamma correction in different channels the gamma is also chosen at random in the range(0.5,1.5).Another data augmentation method that we tried is applying Gaussian noise in the images.We used a truncated normal distribution with standard deviation of 0.01 and a mean of 0 to generate these random values which we then added to the different channels of the image.In the implementation they have used horizontal flip we tried vertical flipping but it did not have much effect hence we removed it from our implementation.

E. Cheirality Check

Generally for chierality condition we check that both the camera should have more numeber positive depth points when w.r.t both the camera frame references. i.e. maximize no of count of $r_3 \times (X - C) > 0$ and Z¿0. We use this as a loss function. From the predicted depth we calculate the dot product of extrinsic matrix and predicted depth take the last element and make sure the number of positive depth are more. IT is shown as follows

$$A = P_3.X \tag{2}$$

where P_3 is the last row of extrinsic matrix.

$$A_{pos} = max(0, A) \tag{3}$$

Above equation is RelU.

$$A_{pos}^N = \sum_{max} \frac{A_{pos}^2}{A_{pos}^2 + \epsilon} \tag{4}$$

Where epsilon is a small value, which is used to tackle division by zero ($\epsilon = 0.0000002$)

$$L_{chirality} = \frac{1}{A_{pos}^N} \tag{5}$$

We have added this loss to the other loss present in the SFMLearner with α=0.8.

TABLE I: Error Comparison

SfM Learner Model	abs_diff	abs_rel	sq_rel	rms	log_rms	d1_all	a1	a2	a3
Baseline	5.8204	0.3830	5.6940	8.9870	0.4780	0.3932	0.4473	0.7174	0.8351
Data Augmentation and varying the Learning Rate	9.1403	0.5038	7.2140	14.3478	0.9772	0.7589	0.1934	0.3917	0.4925
Epipolar Constraint	5.7893	0.3560	3.6732	8.817	0.4964	0.4097	0.4224	0.6182	0.8279
SSIM	5.8591	0.3790	3.7123	8.8950	0.5190	0.4289	0.4345	0.6345	0.8345
Cheirality	5.5812	0.3645	3.6461	8.8563	0.5086	0.4188	0.4280	0.6211	0.8298

IV. Training

All of the training is done on a system with an i7-11800H and an RTX3060 and RTX3070. Except for the Epipolar constraint, which was only trained for 20 epochs due to time constraints, all implementations are trained for 200 epochs.It took about 15 hours to train on 200 epochs.

V. Results

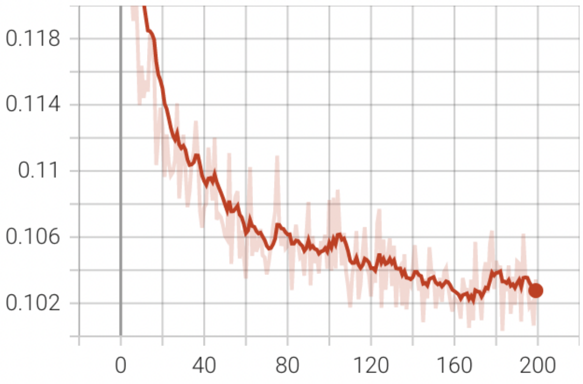


Fig. 3: Validation total loss for Baseline Model for 200 epochs

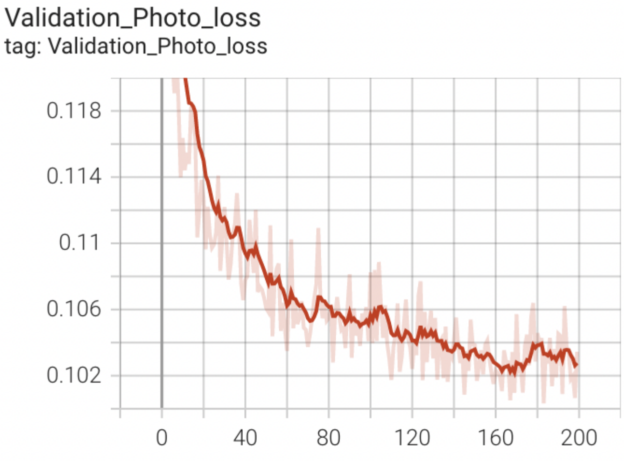


Fig. 4: Validation photometric loss for Baseline Model for 200 epochs

VI. Conclusion

From the experiments that we performed by modifying the existing model we conclude that epipolar constraint performed quite well as compared to other modifications that

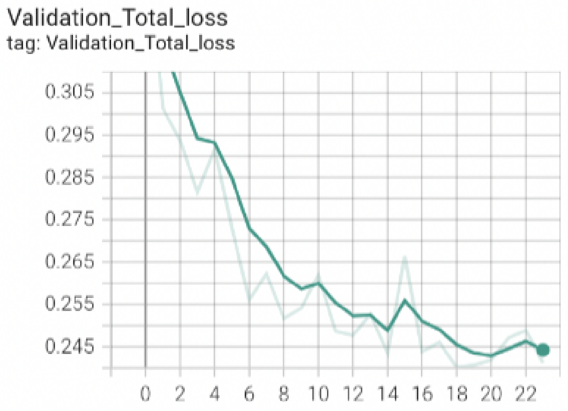


Fig. 5: Validation total loss for epipolar constraint for 20 epochs

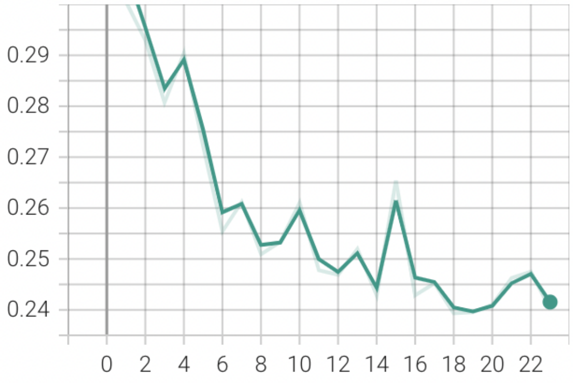


Fig. 6: Validation photometric loss for epipolar constraint for 20 epochs

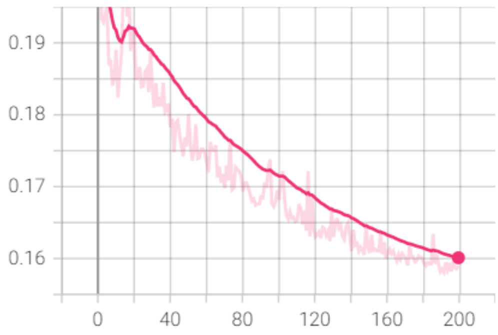


Fig. 7: Validation total loss for data augmentation for 200 epochs

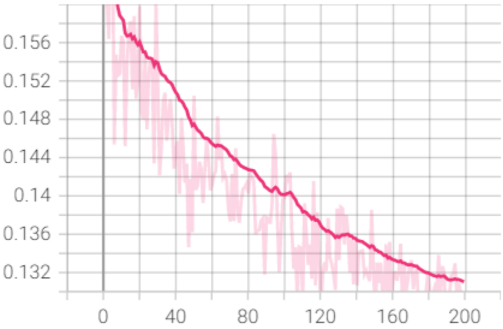


Fig. 8: Photometric loss for data augmentation for 200 epochs

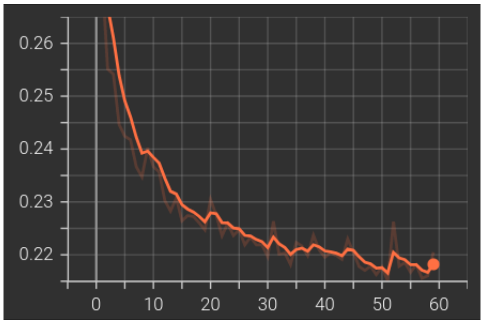


Fig. 9: Validation total loss for cheirality modification for 60 epochs

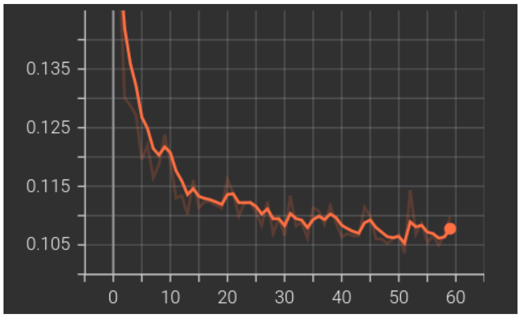


Fig. 10: Validation Photometric loss for cheirality modification for 60 epochs

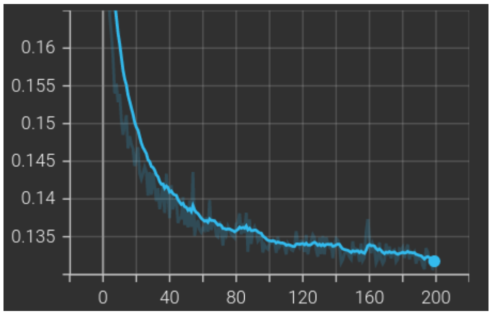


Fig. 11: Validation total loss for SSIM for 200 epochs

we did.Other modifications like cheirality also performed well and it gave almost same output as compared to the epipolar output.Even after augmenting the data using different methods it did not give the expected output and it performed very badly as compared to other modifications.We think that after increasing the brightness due to high brightness the image got saturated and due to which it may have failed to get enough features for calculating the depth and disparity value.In the future versions we try to overcome this shortfalls and we will try to integrate all the modifications in a single network to get much better results than the ground truth.

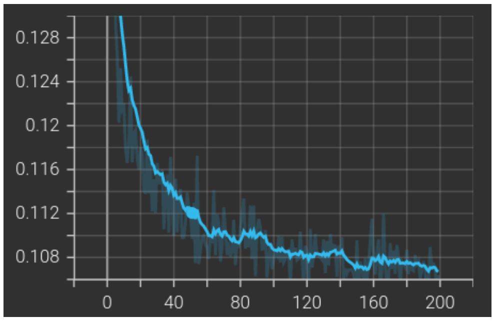


Fig. 12: Validation Photometric loss for SSIM for 200 epochs

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

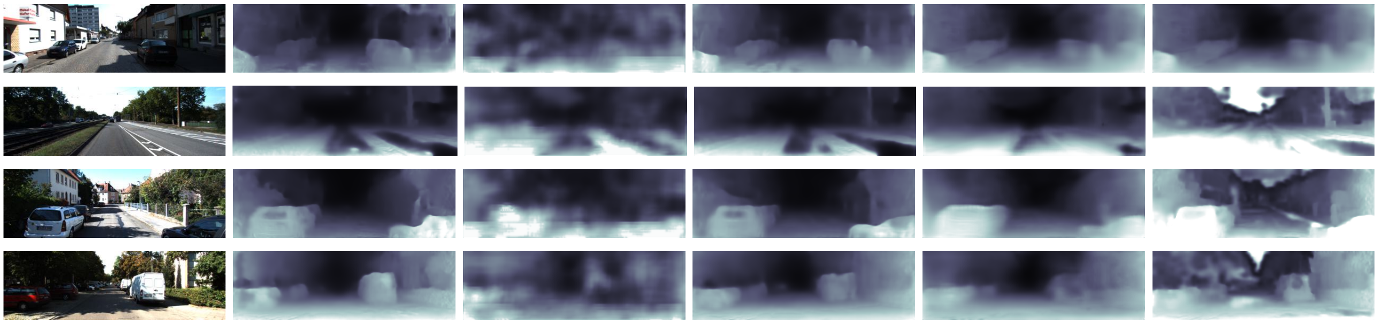
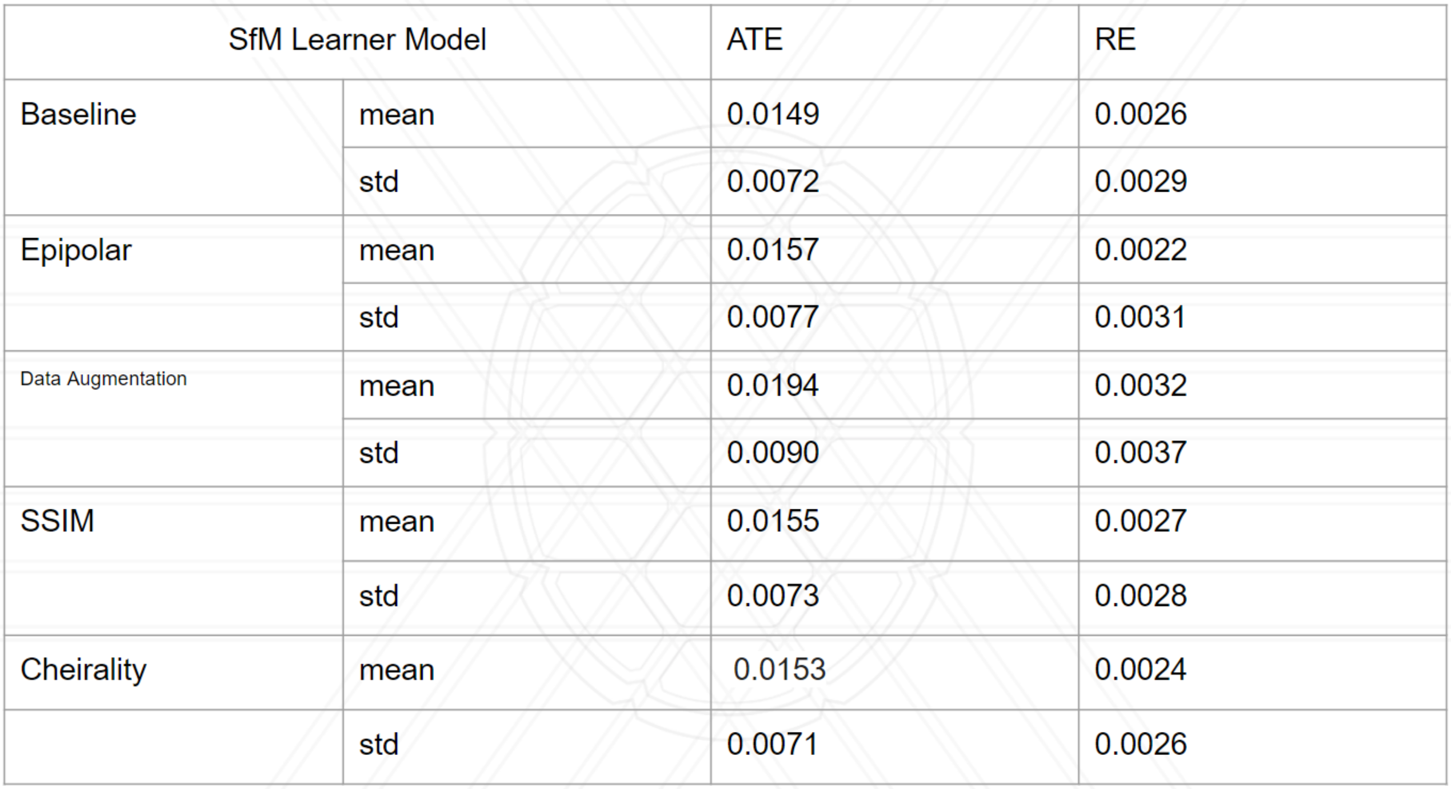


Fig. 13: 1: Input image, 2: Baseline, 3: Augmentation, 4: Cheirality, 5: Epipolar, 6: SSIM



SfM Learner Model		ATE	RE
Baseline	mean	0.0149	0.0026
	std	0.0072	0.0029
Epipolar	mean	0.0157	0.0022
	std	0.0077	0.0031
Data Augmentation	mean	0.0194	0.0032
	std	0.0090	0.0037
SSIM	mean	0.0155	0.0027
	std	0.0073	0.0028
Cheirality	mean	0.0153	0.0024
	std	0.0071	0.0026

Fig. 14: Pose error comparison between original approach and our Implementation for seq 4 and seq 7