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

Rohith Jayarajan (115458437)
University of Maryland
College Park, Maryland 20740
rohith23@umd.edu

Rohitkrishna Nambiar (115507944)
University of Maryland
College Park, Maryland 20740
rohit517@umd.edu

Srinidhi Sreenath (115526723)
University of Maryland
College Park, Maryland 20740
ssreenat@terpmail.umd.edu

***Abstract*—In the previous project, we dealt with reconstructing 3D structure of a given scene from multiple views using the traditional approach pipeline for Structure-from-Motion. In this project, we explore the deep learning approach to reconstruct the 3D scene by learning the methods used in the SFMLearner paper by David Lowes team at Google [1]. This presents the approach for the task of monocular depth estimation and camera motion estimation from unstructured video sequences. Also, tweaking of the network is studied to make it perform better.**

I. Introduction

The goal of the SfMlearner is to learn the ego-motion by mimicking the way humans learn to infer ego motion and scene structures over time. It is difficult to recreate an interpretation of the geometry and motion in a real world scene. Years of research in geometric computer vision has not been able to model the scene and motion as good as a human.

One way of approaching this problem would be to learn the way a human would. One approach is to train a model that observes sequences of images and aims to explain its observations by predicting likely camera motion and the scene structure i.e estimating depth given an image and transformation given a set of images. The transformation represents rotation and translation (a total of 6 DoFs) and the depth map is a per pixel estimate of ego-motion. The approach is end to end and unsupervised i.e given an input set of sequence of images, the depth map is estimated for each image and pose estimated between subsequent images in time.

The unsupervised approach means no labeling of data and no camera motion information is required. The approach is inspired by *scene synthesis*, which performs consistently well when the convolution neural network is trained on diverse layout and appearance structures. By training the network to learn scene synthesis, it also implicitly learns to estimate depth per image and pose between 2 images. Datasets like KITTI can be used to empirically evaluate the model and demonstrate effectiveness.

II. Approach

In the SFMLearner paper, the authors propose a Convolutional Neural Network framework to jointly train the single view depth and the camera pose estimate from unlabeled video sequences. So this is an unsupervised learning approach to solve the problem for monocular depth estimation and pose estimation. But an interesting feature of this CNN framework is that the outputs from the model can be independently used in inference during test time. The major assumption is that the scenes in the video scenes are rigid, i.e, ego motion is not considered.

Given one image of the scene, a target view of that image is synthesized with given the pose, per-pixel depth and the visibility in a nearby view. This view synthesis is done in a fully differentiable manner and the objective is formulated as shown in equation 1

$$L = \sum_{s}\sum_{p} |I_t(p) - \hat{I}_s(p)| \tag{1}$$

where p indexes over the pixel coordinated and $\hat{I}_s$ is the source view I_s warped to the target coordinate frame. This can be applied to standard videos without the information of the pose estimate.

As equation 1 indicates, the target frame I_t is reconstructed by sampling pixels in I_s based on the depth map $\hat{D}_t$ and relative pose $\hat{T}_{t\to s}$. If p_t is the pixel position in target view, and K is camera intrinsic matrix, from equation , p_t's projected coordinates onto source view p_s can be obtained and then the value of $I_s(p_s)$ is linearly interpolating since p_s is continuous. The interpolated value is then used to fill $\hat{I}_s(p_t)$.

$$p_s \sim K\hat{T}_{t\to s}\hat{D}_t(p_t)K^{-1}p_t \tag{2}$$

The entire procedure is accomplished with the implementation of *spatial transformer networks*.

Model limitations: Since the error calculation is based on view synthesis from monocular videos, the following assumptions hold:

- The scene is mostly static i.e no dynamic objects.
- There are no occlusions between the source and target frames.

If these assumptions are not held during training then the model won't be perfect. To be robust to these assumptions during the training phase, an *explainability prediction network* along with depth and pose networks is used to obtain a per pixel soft max for each target source pair indicating networks belief that the model holds the above assumptions. To avoid overfitting the model, the *regularization term* is used that resolves the trivial solution to the network's error.

A well known problem in motion estimation is that the gradients are derived from the pixel intensity differences between I and its four neighbors. The two strategies used in the SFMLearner paper are:

1) Using an encoder-decoder based convolutional architecture with a small bottleneck for depth network.
2) Explicit multi-scale and smoothness loss that allows gradients to be derived from larger spatial regions directly

A. Network Architecture

1) For monocular single view depth prediction, the authors of SFMLearner used the architectures of that in DispNet. It is based on an encoder decoder design with skip connections.
2) For pose estimation, the input is the target view images stacked with the source view which then is used to aggregate predictions at all spatial locations.
3) An explainability prediction network is also used which shares with the pose network the first five feature encoding layers. All the convolution and deconvolution layers use the ReLU activation function and the prediction layers use no activation function.

B. Modifications in Architecture

The changes made to the network are as follows:

1) The loss function was changed to L_2 loss from L_1 loss.
2) The number of filters in layer *cnv1*, *cnv2*, *cnv3*, *cnv4*, and *cnv5* were changed to 32, 64, 128, 256, and 512 respectively.
3) The number of filters in *cnv6*, *cnv7* were set to 512.
4) The number of filters in *upcnv5*, *upcnv4*, *upcnv3*, *upcnv2*, and *upcnv1* were changed to 512, 256, 128, 64, and 32 respectively.

III. Challenges

We faced the following challenges during the course of the project:

- The training of models was done on google cloud platform. The model needed to be trained for 100k iterations to get good results. We could successfully train for 24000 iterations as the google cloud platform lost connection frequently and we had to restart the training from scratch.
- **Tensorboard plots could not be obtained because the training was done on the google cloud platform.**

IV. Results

A. Depth Comparison

Following are the comparison of depth estimation outputs obtained by the original SfMLearner model and the model trained by us.

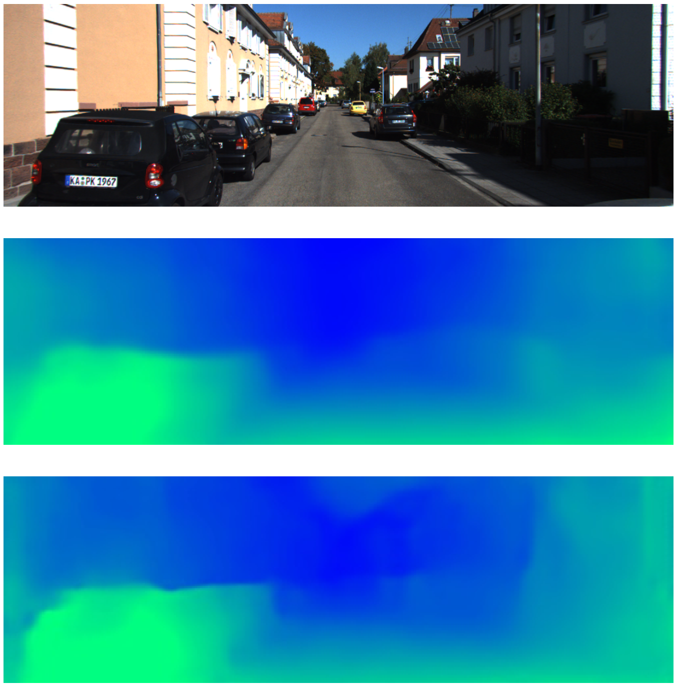

Fig. 1: (a) RGB frame of the image sequence. (b) Output from SFMLearner paper architecture (a) Output from our modified version of SFMLearner.

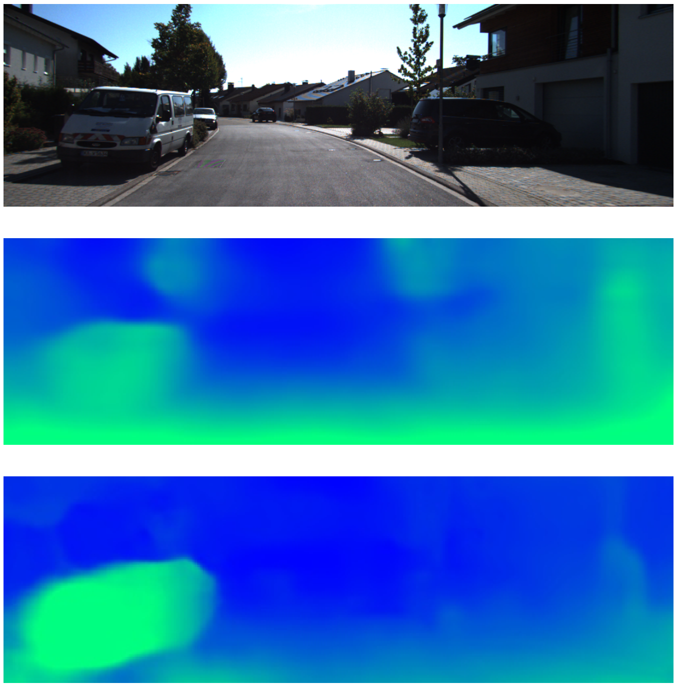

Fig. 2: (a) RGB frame of the image sequence. (b) Output from SFMLearner paper architecture (a) Output from our modified version of SFMLearner.

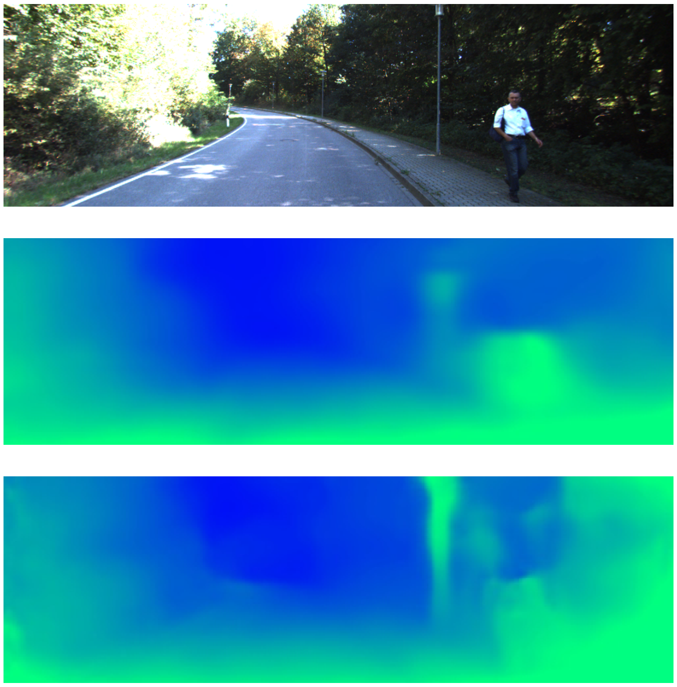

Fig. 3: (a) RGB frame of the image sequence. (b) Output from SFMLearner paper architecture (a) Output from our modified version of SFMLearner.

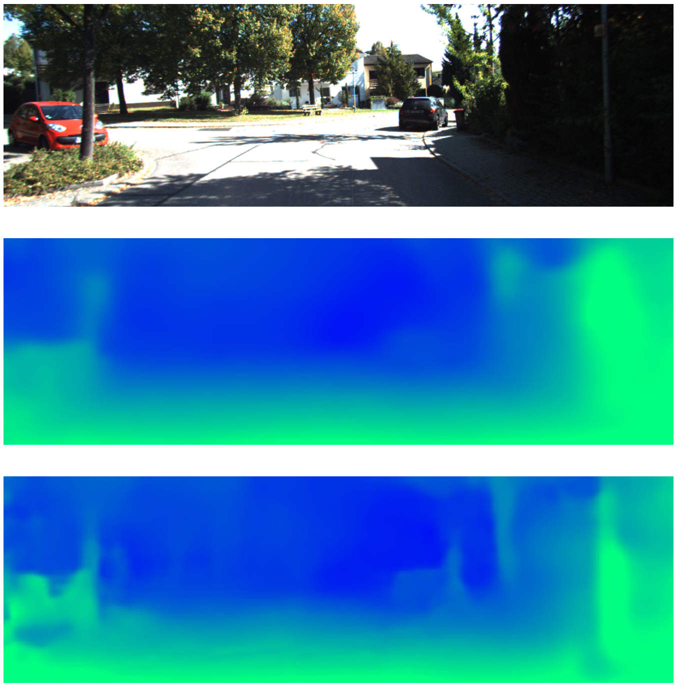

Fig. 4: (a) RGB frame of the image sequence. (b) Output from SFMLearner paper architecture (a) Output from our modified version of SFMLearner.

B. Odometry Comparison of Pose Estimation

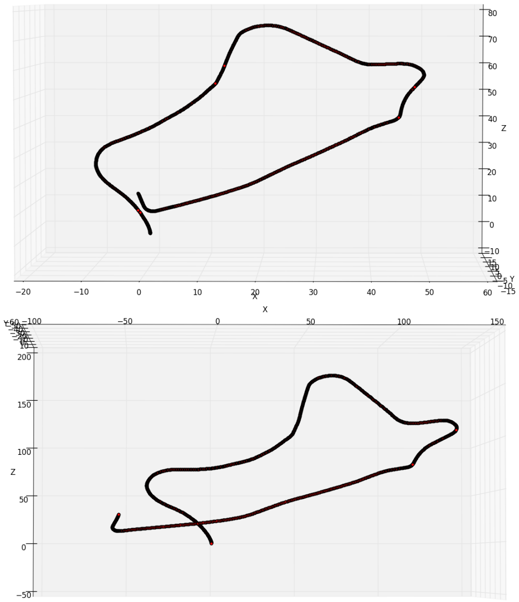

Fig. 5: (a) Trajectory plot using modified architecture for Test sequence 9 (b) Trajectory plot using SfMLearner architecture for Test sequence 9.

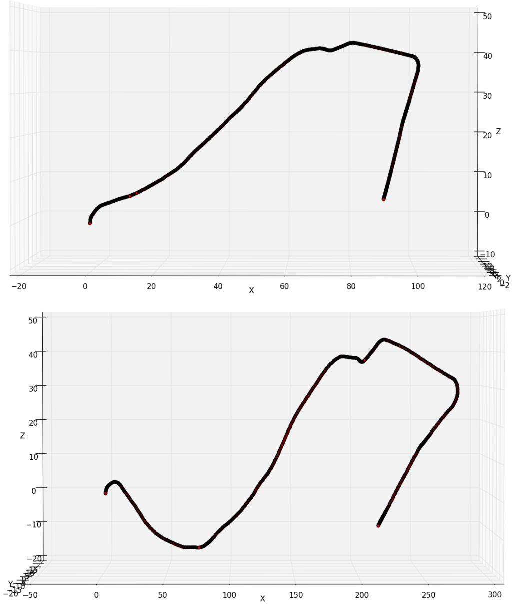

Fig. 6: (a) Trajectory plot using modified architecture for Test sequence 10 (b) Trajectory plot using SfMLearner architecture for Test sequence 10.

References

[1] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” *CoRR*, vol. abs/1704.07813, 2017. [Online]. Available: http://arxiv.org/abs/1704.07813