Deep Fundamental Matrix Estimation*

Guandao Yang gy46@cornell.edu
Omid Poursaeed op63@cornell.edu
Hanqing Jiang hj284@cornell.edu
Qiuren Fang qf32@cornell.edu
Bharath Hariharan bharathh@cs.cornell.edu
Serge Belongie sjb344@cornell.edu

Abstract

Estimating fundamental matrices is a classic computer vision problem. Traditional methods rely strongly on the correctness of the key-point correspondences, which can be noisy and unreliable. As a result, these methods find it difficult to handle image pairs with large occlusion or significantly different camera poses. In this report, we design a ConvNet architecture to estimate the Fundamental Matrices. Our model can be trained in an end-to-end fashion without key-point correspondences. We analyze performance of the proposed model using various metrics. We also conduct an ablation study to examine effectiveness of different components of the model.

1 Introduction

The fundamental matrix (F-matrix) contains rich information between two stereo images, including relative camera intrinsic, rotation, and translation. The ability to estimate the fundamental matrix is essential for many computer vision applications such as camera calibration, camera location, depth estimation, 3D reconstruction, etc. The popular approach to this problem is based on detecting and matching local feature points, then using the obtained correspondences to compute the fundamental matrix by solving an optimization problem about the epipolar constraints [1, 2].

The performance of such methods is highly dependent on the accuracy of the local feature matches, which are based on algorithms like SIFT [3]. These methods, however, are not always reliable. For instance, feature matching based on SIFT could not well handle large occlusion, large translation, or large rotation between two scenes.

In order to reduce how much we rely on key-point correspondences to estimate the F-matrix, we propose an end-to-end trainable method that does not rely on key-point correspondences using deep learning. In Sec. 3, we will present our detailed network architecture. The main challenge of using deep learning to directly regress the F-matrix is to preserve its mathematical properties. We designed a reconstruction module (Sec. 3.1) and a normalization layer (Sec. 3.3) to address these two challenges. Finally, empirical experiments on a synthetic dataset are shown in Sec 4.

2 Related Works

Geometry Methods: Estimation of fundamental matrix goes back to the eight-point algorithm proposed by Longuet-Higgins [1] and optimized by Hartley [4]. Later, people used RANSAC [5] to find inliers and get a more robust estimation. These methods minimize the re-projection error, Sampson distance and other loss functions. Key-point correspondences are mostly computed from hand-crafted feature extractors like SIFT [3]. These models tend to fail in the case where viewpoints

*This project started as Guandao Yang's independent study with Omid Poursaeed during the 2016 fall semester. The independent study was advised by Professor Serge Belongie. Then it was continued as a class project in Professor Bharath Hariharan's CS 6670 class with Hanqing Jiang and Qiuren Fang during 2017 fall semester.

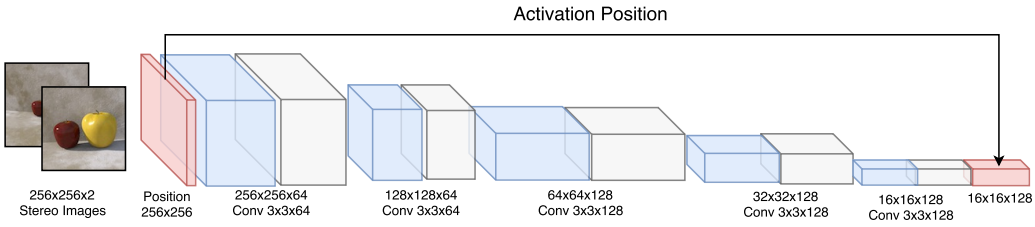

Figure 1: Network Architecture. A Convolutional Neural Network is used as the feature extractor.

are drastically different or where higher-level features are needed to make correct correspondence decisions.

Deep Learning: As shown in recent papers such as HomographyNet [6] and [7], ConvNets could be used to directly regress the Homography matrix. This report is inspired by the HomographyNet paper and tries to adapt the idea to estimate Fundamental matrices. People have successfully applied deep learning to estimate the camera position [8] and to perform Structure from Motion [9]. The success of these works shows that ConvNets are capable of computing information needed for estimating the F-matrices. Our work tries to completely recover the F-matrices.

3 Network Architecture

Our network architecture is inspired by HomographyNet [6]. We use a VGG-like architecture since such ConvNets are good at extracting higher-order features. These features can be highly useful to handle cases where large occlusion, translations, or rotations appear between the stereo images. Figure 1 illustrates our network structure.

The final image features will be used in two ways: 1) put into a MLP to produce 9 parameters, and these 9 parameters will be put into a normalization layer and output a normalized matrix as the prediction; and 2) put into a MLP to predict 8 parameters, and these 8 parameters will be used to reconstruct a F-matrix using the reconstruction layer and the normalization layer. The reconstruction layer will be presented in Sec. 3.1, and the normalization layer will be presented in Sec. 3.3.

3.1 F-matrix Reconstruction Layer

A main challenge to directly regress all the entries of the Fundamental matrices is that the predicted matrix might not satisfy all the mathematical properties required for a fundamental matrix. For example, F-matrix is a rank-2 matrix with seven degrees of freedom. These two properties cannot be enforced by a model that directly regresses the nine matrix entries.

To address this problem, we leverage the following formula to reconstruct the fundamental matrix [2] [1]:

$$\hat{\mathbf{F}} = {\mathbf{K_2}}^{-1}\mathbf{R}[\mathbf{t}]_\times {\mathbf{K_1}}^{-1} \tag{1}$$

where $\mathbf{K_2}, \mathbf{K_1}$ are camera intrinsics, $\mathbf{R}$ is the relative camera rotation, and $[\mathbf{t}]_\times$ is the relative camera translation. Note that these four matrices can be constructed by eight parameters (f_1, f_2, r_x, r_y, r_z, t_x, t_y, and t_z) in the following way:

$$\mathbf{K}_i = \begin{bmatrix} f_i^{-1} & 0 & 0 \\ 0 & f_i^{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix}, [\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_x & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}, \mathbf{R} = \mathbf{R_x}(r_x)\mathbf{R_y}(r_y)\mathbf{R_z}(r_z) \tag{2}$$

where $\mathbf{R_x}$, $\mathbf{R_y}$, and $\mathbf{R_z}$ are 3D rotation matrices around x, y, and z-axis respectively. Note that the predicted $\hat{\mathbf{F}}$ is differentiable with respect to these eight parameters. So we can construct a layer f_r that takes the eight parameters and outputs a fundamental matrix $\hat{\mathbf{F}}$ to enforce the mathematical properties required by a fundamental matrix, such as being rank-2 and having 7 degrees of freedom.

[1] We assume $u_0 = v_0 = 0$.

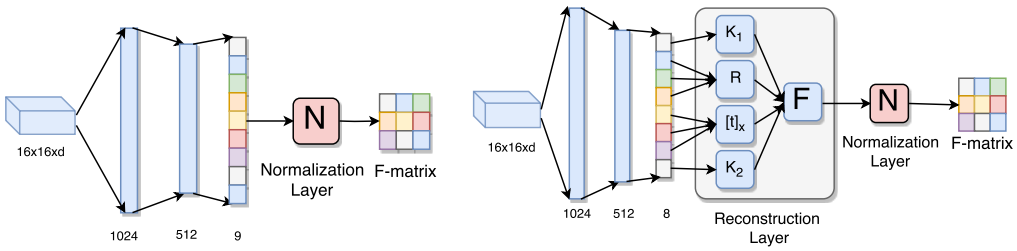

Figure 2: Two ways to predict F-matrix using image features. 2(a) (left) is the architecture to directly regress the entries of the F-matrix. 2(b)(right) illustrates the reconstruction layer discussed in Sec. 3.1. Both need to normalize the F-matrix before comparing with the ground truth to compute the losses.

3.2 Location Aware Max-pooling

The other challenge comes from the fact that the features extracted by the ConvNet are agnostic to locations, especially when these features are down-sampled in the max-pooling layer. The locations of these activation, however, are essential for computing the F-matrix. Therefore, in the max-pooling layer, we keep all the indices of where the activations come from. At the end of the ConvNet, we append the position of where the final features come from with respect to the full-size image. Each location is indexed with an integer from $[0, 65535]$ divided by 65535 to be normalized within range $[0, 1]$. At the end, each feature will have a position from which it comes from. As a result, the output feature vector will double its depth from $16 \times 16 \times 128$ to $16 \times 16 \times 256$.

3.3 Normalization

In order to make the predicted F-matrix comparable with the ground-truth, we have to eliminate the scale factor, since if we multiply the F-matrix by any none-zero scalar, the F-matrix will remain valid. The common practice is to divide the F-matrix by its last entry. We call this method **ETR-norm**. However, this could result in some large entries since the last entry of the F-matrix could be close to zero. With these large entries, the training becomes unstable. As a result, we propose two alternative normalization methods.

FBN-norm: We divide all entries of a F-matrix by its Frobenius norm, so that all matrices live on a 9-dimensional sphere. Let $\|\mathbf{F}\|_F$ denote the Frobenius norm of matrix $\mathbf{F}$. Then:

$$\mathcal{N}_{FBN}(\mathbf{F}) = \|\mathbf{F}\|_F^{-1}\mathbf{F} \tag{3}$$

ABS-Norm: We divide all entries of a F-matrix by its maximum absolute values, so that all entries of the matrix will be restricted within $[-1, 1]$:

$$\mathcal{N}_{ABS}(\mathbf{F}) = (\max_{i,j} |\mathbf{F}_{i,j}|)^{-1}\mathbf{F} \tag{4}$$

During training, the normalized F-matrices are compared with the ground-truth with both L_1 and L_2 losses. We provide empirical results to study how each of these normalization methods will influence the performance and the stability of training in Sec. 4.

4 Experiments

To evaluate whether the design of our model could successfully learn the manifold of F-matrices, we train four models with different configurations and compare their performance on metrics defined in Sec. 4.2. The baseline model (**Base**) uses no position features, nor does it use the reconstruction module. The **REC** model is the same as **Base** but uses reconstruction module. The **POS** model adds on top of the **Base** model and utilizes the position feature. Finally, the **REC+POS** model uses both the position feature and the reconstruction module. Comparison results are shown in Table 1.

4.1 Dataset

In order to obtain ground-truth F-matrices for training, we develop a synthetic dataset based on POV-Ray [10]. We use a simple scene to render roughly 600 different images, all of which could see

Metrics	Models	ETR	FBN	ABS	Metrics	Models	ETR	FBN	ABS
EPI-SQR	Base	1610.57	597.45	254.72	SSD	Base	30356.01	29261.65	22646.30
	REC	2094.55	**1.15**	29.40		REC	101694.60	99434.52	87974.36
	POS	272.12	420.30	353.82		POS	29394.36	22582.73	38256.89
	REC+POS	**139.43**	6.90	**27.85**		REC+POS	95064.84	97621.61	>1e5
	ground truth	9941.73	172.62	227.88		ground truth	**483.27**	**487.74**	**496.66**
EPI-ABS	Base	27.93	18.03	11.77	SED	Base	34.00	4.97	1.33
	REC	2.44	**1.02**	**1.30**		REC	7848.15	**2e-5**	2.15
	POS	9.57	15.95	14.52		POS	**5.54**	3.10	**2.03**
	REC+POS	**2.00**	1.16	1.39		REC+POS	78.79	0.04	4.93
	ground truth	10.86	3.71	4.42		ground truth	>3e7	2721.04	4561.39

Table 1: Evaluation metrics for different models and normalization methods. For most of the metrics, our method could achieve better precision compared to the F-matrix ground truth it was trained on.

the center of the scene. Then we pair up all the images and compute the ground truth Fundamental matrices and key-point correspondences between two images using OpenCV [11]. Note that only a small part of key-point correspondences are used to compute the ground-truth using either seven-points or eight-points algorithm, and the remaining points are held out for evaluation.

4.2 Evaluation Metrics

To evaluate both the ground truth and the predicted F-matrices, we use four metrics, all of which measure how well the F-matrix satisfies the epipolar constraint according to the held out key-points.

EPI-ABS (Epipolar Constraint with Absolute Value):

$$\mathcal{M}_{EPI-ABS}(\mathbf{F}, \mathbf{p}, \mathbf{q}) = \sum_i |\mathbf{p}_i^T \mathbf{F} \mathbf{q}_i| \tag{5}$$

EPI-SQR (Epipolar Constraint with Squared Value):

$$\mathcal{M}_{EPI-SQR}(\mathbf{F}, \mathbf{p}, \mathbf{q}) = \sum_i (\mathbf{p}_i^T \mathbf{F} \mathbf{q}_i)^2 \tag{6}$$

SSD (Sampson Distance):

$$\mathcal{M}_{SSD}(\mathbf{F}, \mathbf{p}, \mathbf{q}) = \sum_i \frac{(\mathbf{p}_i^T \mathbf{F} \mathbf{q}_i)^2}{(\mathbf{F}\mathbf{p}_i)_1^2 + (\mathbf{F}\mathbf{p}_i)_2^2 + (\mathbf{F}\mathbf{q}_i)_1^2 + (\mathbf{F}\mathbf{q}_i)_2^2} \tag{7}$$

SED (Symmetrical Epipolar Distance):

$$\mathcal{M}_{SED}(\mathbf{F}, \mathbf{p}, \mathbf{q}) = (\mathbf{p}_i^T \mathbf{F} \mathbf{q}_i)^2 \Big(\frac{1}{(\mathbf{F}\mathbf{p}_i)_1^2 + (\mathbf{F}\mathbf{p}_i)_2^2} + \frac{1}{(\mathbf{F}\mathbf{q}_i)_1^2 + (\mathbf{F}\mathbf{q}_i)_2^2} \Big) \tag{8}$$

We refer readers to [2] which discusses pros and cons of each metric. We evaluate our models on all four metrics. The results are shown in Table 1 and discussed in Sec. 4.3.

4.3 Results and Discussion

Results are shown in Table 1. The prediction outputted from the neural network is significantly better than ground truth generated from OpenCV in terms of **EPI-ABS**, **EPI-SQR** and **SED** metrics. However, the **SSD** (Sampson Distance) metric for the prediction is consistently higher compared to the ground truth.We are still investigating the source of this phenomenon.

Adding the reconstruction module significantly reduces the error no matter which kind of feature extractor it is based on. This shows the effectiveness of the reconstruction module. But adding the position feature does not show consistent improvement. Our hypothesis is that since right now the stereo-images pairs are passed into the network by channels, the position features might not provide meaningful gain in this setting. A future work will be to use Siamese structure and extract position features separately from each image.

It is worth noting that the large error in OpenCV ground truth can be caused by the artifact from the synthetic images. Usually, the metrics **EPI-SQR** should get a value of less than 1 in order to be considered to be a good F-matrix. Therefore, another direction for future work will be to improve the quality of the dataset or run on other real-world datasets such as KITTI [12].

References

[1] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.

[2] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[3] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[4] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997.

[5] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016.

[7] Syed Ammar Abbas et al. Recovering homography from camera captured documents using convolutional neural networks. *arXiv preprint arXiv:1709.03524*, 2017.

[8] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[9] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. *arXiv preprint arXiv:1612.02401*, 2016.

[10] Tomas Plachetka. Pov ray: persistence of vision parallel raytracer. In *Proc. of Spring Conf. on Computer Graphics, Budmerice, Slovakia*, pages 123–129, 1998.

[11] Itseez. Open source computer vision library. `https://github.com/itseez/opencv`, 2015.

[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.