DEEP FUNDAMENTAL MATRIX ESTIMATION

REN’E RANFTL AND VLADLEN KOLTUN

HAMMAD AYYUBI
CSE-291D
Contents

- Background and Topic Summarization
  - What is Fundamental Matrix?
  - Examples
  - Estimation and main challenges

- Related Work
  - Key ideas and their shortcomings

- Technical Details
  - Preliminaries
  - Deep Model Fitting
  - Fundamental Matrix Estimation

- Experiments
  - Tanks and Temples
  - KITTI
  - Unstructured SFM
  - Homography

- Critical Review
  - Strengths
  - Weakness

- Extensions and Follow-ups
BACKGROUND AND TOPIC SUMMARIZATION
WHAT IS FUNDAMENTAL MATRIX?

Epipoles and Epipolar Lines

Epipole: location of cam2 as seen by cam1.

Epipole: location of cam1 as seen by cam2.
For point \( x \), Fundamental Matrix (\( F \)) relates it to the epipolar line \( l' \):

\[
l' = Fx
\]
Key Characteristics of Fundamental Matrix

- It's a 3x3 matrix
- It has rank 2
- It depends on both extrinsic (R&T) and intrinsic parameters (K) of camera
- Differs from essential matrix in the fact that essential matrix only depends on external parameters
Background And Topic Summarization

Examples

CRCF, UCF
Background And Topic Summarization

Fundamental Matrix Estimation

Eight Point Algorithm

\[
\begin{bmatrix}
    x'_i & y'_i & 1 \\
    x_i & y_i & 1
\end{bmatrix}
\begin{bmatrix}
    f_{11} & f_{12} & f_{13} \\
    f_{21} & f_{22} & f_{23} \\
    f_{31} & f_{32} & f_{33}
\end{bmatrix}
\begin{bmatrix}
    x'_i \\
    y_i \\
    1
\end{bmatrix} = 0
\]

\[
\begin{bmatrix}
    x_1' & x_1y' & x_1 & y_1' & y_1 & x'_1 & y'_1 & 1 \\
    x_n' & x_ny' & x_n & y_n' & y_n & x'_n & y'_n & 1
\end{bmatrix}
\begin{bmatrix}
    f_{11} \\
    f_{21} \\
    f_{31} \\
    f_{12} \\
    f_{22} \\
    f_{32} \\
    f_{13} \\
    f_{23}
\end{bmatrix} = 0
\]

\[
Ax = 0
\]

\[
\min_x ||Ax||^2 \text{ s.t. } ||x||^2 = 1
\]

Solution:
- Eigenvector of A with the smallest eigenvalue
- Enforce rank 2 of A by using SVD
Main Challenges in Fundamental Matrix Estimation

- Presence of Outliers/imperfect correspondences
- Different noise distribution in Inliers (Gaussian Assumption in most)
- Huge solution space (Can we narrow it down?)
RELATED WORK
The idea is to find a geometric model that has the most support in terms of inliers. Inliers are defined on the basis of some distance metric from ground truth and a threshold.

Generic steps:
1. A set of points are sampled and a non-robust model is estimated from it.
2. The model is then scored by evaluating a scoring function on all points. The current model is accepted if it’s score is better than all previous models.
3. Iterate until a stopping criterion is reached

Shortcoming:
The biggest issue is that as the minimal number of data points required to correctly estimate the model increases, the probability of sampling outlier increases exponentially. For example if half of the correspondences are spurious, the probability of picking up correct ones is only $0.5^{28} = 0.39\%$.
Global Optimizations

Yang et al. and Li et al. demonstrate work in which the goal is also achieving consensus set maximaization. But they try to achieve it by optimizing globally. Essentially, pose the problem as Integer Programming Problem.

Shortcoming:
- Since the underlying problem is NP-Hard the process is slow, at times degrading to exhaustive search.

Robust base estimators

Robustly estimate base models using a series of weighted least squares problem (Hoseinnezhad et al.).

Shortcomings:
- Careful initialization of weight parameters
- Gaussian Noise assumption in inliers

Hoseinnezhad, R., Bab-Hadiashar, A.: An M-estimator for high breakdown robust estimation in computer vision. Computer Vision and Image Understanding
TECHNICAL DETAILS
Many geometric problems can be reduced to a least square fitting problem. Given a set of points \( P \in \mathbb{R}^{N \times D} \), model parameters \( x \in \mathbb{R}^D \), the problem can be modelled as:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{N} \| (A(P))_i \cdot x \|^2 \\
\text{subject to} & \quad \| x \| = 1,
\end{align*}
\]

\[
A : \mathcal{P} \rightarrow \mathbb{R}^{kN \times d'}
\]

- An example could be fitting a plane given a set of 3D points.
Technical Details

**Challenge:** Presence of outliers. The estimated model is off depending on the number of outliers.

**Solution:** One possible way is to introduce a weighting function which gives low priority to outliers.

\[
x^{j+1} = \arg \min_{x: \|x\|=1} \sum_{i=1}^{N} w(p_i, x^j) \| (A(P))_i \cdot x \|^2
\]

- The equation doesn’t adhere to a closed form in general, thus the solution is arrived at by successively reweighing and estimating the model parameters.
- The problem now is to find the correct weight which can take care of outliers.
This work is also based on weighted least squares problem.

Aims to learn the weighing function/parameter $w$.

The method is able to perform better in cases where one or more of following is correct:

1. The input data admits regularity in inlier and outlier distribution. Ex: Uniform outlier distribution.
2. Extra information which can be leveraged to further re-weigh the outliers. Ex: Keypoint geometry information.
3. The output space is limited to a narrow region of solution space. Ex: Driving dataset
**Technical Details**

**Architecture**

Basic building blocks:
1. Model Estimator
2. Weight Estimator
Technical Details

Model Estimator

From our least square weighted equation:

\[ x^{j+1} = \arg \min_{x: \|x\|=1} \sum_{i=1}^{N} w(p_i, x^j) \| (A(P))_i \cdot x \|^2 \]

We redefine \( w \) according to our deep network – \( w : P \times S \times \mathbb{R}^{d'} \to (\mathbb{R}_{>0})^N \)
where, \( S = R^{N \times s} \) collects side information for each point.

Now, our least square weighted equation becomes:

\[ x^{j+1} = \arg \min_{x: \|x\|=1} \sum_{i=1}^{N} (w(P, S, x^j; \theta))_i \| (A(P))_i \cdot x \|^2 \]

\[ x^{j+1} = \arg \min_{x: \|x\|=1} \| W^j(\theta)Ax \|^2 \]
**Model Estimator**

\[ x^{j+1} = \arg \min_{x: \|x\|=1} \left\| W^j(\theta)Ax \right\|^2 \]

Solution:

- Eigenvector \( v_{a'} \) of the matrix \( W^j(\theta)A \) corresponding to the smallest eigen value.
- One step left – enforcing rank 2 of fundamental matrix. It can be achieved by considering a function \( g(x) \) which maps the eigenvector \( v_{a'} \) to the parameters of the model. (Fundamental Matrix)
- Concisely, we get the model parameters from the matrix \( W^j(\theta)A \) by:
  \[ g(f(W(\theta)A)) \text{, where } f(X) = v_{a'} \]
- For training end to end, we need \( g(x) \)and \( f(X) \) to be differentiable.

\[
\frac{\partial g}{\partial X} = U \left\{ 2\Sigma \left( K^T \circ \left( V^T \frac{\partial g}{\partial v_{a'}} \right)_{\text{sym}} \right) \right\} V^T,
\]

\[
K_{ij} = \begin{cases} 
\frac{1}{\sigma_i\sigma_j}, & \text{if } i \neq j \\
0, & \text{otherwise}
\end{cases}
\]
Technical Details

Model Estimator

Recap:

- Takes input $P$ and weight $w$ and constructs the matrix $W(\theta)A$
- Applies SVD on $W(\theta)A$ to get the eigenvector $v_d$
- Applies the function $g(x)$ to extract the model parameters (Fundamental Matrix)
Technical Details

Weight Estimator

- The input is a set of points.
- Architecture should be order invariant.
- Based on PointNet

**Technical Details**

**Weight Estimator**

**Input:**
1. Matrix of points $P$
2. Side information $S$ for each point
3. Residuals from the current estimate $r(p_i, g(x^j))$

**Architecture:**
1. A linear layer separately for each point.
2. Leaky ReLu as the non-linear function
3. Instance normalization $\left( I(h) \right)_i = \frac{h_i - \mu(h)}{\sqrt{\sigma^2(h) + \epsilon}}$
   
   where, $h_i$ - feature vector for point $i$
   $\mu(h)$ - mean along dimension $N$
   $\sigma^2(h)$ - variance along dimension $N$
4. Final softmax layer gives the weight $w^j(\theta)_i$
Technical Details

Full Architecture

One Forward Pass:
1. Input weight estimator \( w_{\text{init}}(P, S) \)
2. Repeated application of Estimation Module (Model Estimator + Weight estimator) – optimal 5.
3. Geometric model estimator on the final weights
DEEP FUNDAMENTAL MATRIX

Hammad Ayyubi

One Week Later…
1. What is a fundamental Matrix?
2. Means of Estimation – Eight Point Algorithm
3. Main challenges – outliers
4. Different Methods to address it – RANSAC, Global optimization, IRLS
5. The current work builds of IRLS – tries to estimate weight using deep learning
6. Architecture
RECAP

Full Architecture

One Forward Pass:
1. Input weight estimator
2. Repeated application of Estimation Module (Model Estimator + Weight estimator) – optimal 5.
3. Geometric model estimator on the final weights
Some module definitions need to be specified:
1. Preprocessing step $A(P)$
2. The model extractor $g(x)$
3. The residual $r(p_i, x)$
4. Training Loss
Fundamental Matrix Estimation

The residual $r(p_i, x)$:

$$r(p_i, F) = \left| \hat{p}_i^T F \hat{p}'_i \right| \left( \frac{1}{\|F \hat{p}_i\|_2} + \frac{1}{\|F \hat{p}'_i\|_2} \right)$$

Symmetric epipolar distance

Training Loss:

$$\mathcal{L} = \frac{1}{N_{gt}} \sum_{j=0}^{D} \sum_{i=1}^{N_{at}} \min \left( r(p_i^{gt}, g(x^j)), \gamma \right)$$

$\gamma = 0.5$, ensures hard training data doesn’t dominate the loss.
EXPERIMENTS
• Short Image sequences of Train, Horse, M60 etc. taken from hand held camera.
• Use COLMAP SFM (Johannes et. al) to derive fundamental matrices.
• Use SIFT to find potential correspondences; correspondences which are within 1px of epipolar lines are kept as groundtruth.
TANKS AND TEMPLES

- Experiments with varying depths $D$ of estimation module.
- $D = 5^*$ denotes non-usage of side information
- Direct reg. denotes direct regression from points and correspondences to Fundamental matrix estimation (no geometrical information used)

<table>
<thead>
<tr>
<th></th>
<th>% Inliers</th>
<th>F-score</th>
<th>Mean</th>
<th>Median</th>
<th>Min</th>
<th>Max</th>
<th>Time [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D = 1$</td>
<td>42.30</td>
<td>44.80</td>
<td>3.45</td>
<td>1.00</td>
<td>0.08</td>
<td>1912.67</td>
<td>7</td>
</tr>
<tr>
<td>$D = 3$</td>
<td>44.91</td>
<td>47.25</td>
<td>1.98</td>
<td>0.82</td>
<td>0.08</td>
<td>566.70</td>
<td>18</td>
</tr>
<tr>
<td>$D = 5$</td>
<td>45.02</td>
<td>46.99</td>
<td>2.04</td>
<td>0.83</td>
<td>0.11</td>
<td>285.36</td>
<td>26</td>
</tr>
<tr>
<td>$D = 5^*$</td>
<td>44.60</td>
<td>46.42</td>
<td>2.23</td>
<td>0.84</td>
<td>0.10</td>
<td>391.64</td>
<td>26</td>
</tr>
<tr>
<td>Direct reg.</td>
<td>4.42</td>
<td>9.14</td>
<td>16.67</td>
<td>11.96</td>
<td>0.83</td>
<td>386.15</td>
<td>3</td>
</tr>
</tbody>
</table>

Inliers: Correspondences with epipolar distance < 1 px
F-score: Positives are those correspondences which have epipolar distance from ground truth epipolar line < 1 px
Mean and Median: Mean and median distances to ground truth epipolar matches.
### Experiments

#### Weight Estimator Module

<table>
<thead>
<tr>
<th>Layer</th>
<th># in</th>
<th># out</th>
<th>L-ReLU+IN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>–</td>
<td>64</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>64</td>
<td>128</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>128</td>
<td>1024</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>1024</td>
<td>512</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>512</td>
<td>256</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>256</td>
<td>1</td>
<td>x</td>
</tr>
</tbody>
</table>

#### Direct Regression

<table>
<thead>
<tr>
<th>Layer</th>
<th># in</th>
<th># out</th>
<th>L-ReLU</th>
<th>Instance norm</th>
<th>Max pooling</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>–</td>
<td>64</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>2</td>
<td>64</td>
<td>128</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>3</td>
<td>128</td>
<td>1024</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>4</td>
<td>1024</td>
<td>512</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>5</td>
<td>512</td>
<td>256</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>256</td>
<td>128</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>64</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td>9</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
Experiments

Comparison with other methods:

- Two experiments: with and without ratio test. Ratio test is used in SIFT to increase % of inliers.

<table>
<thead>
<tr>
<th></th>
<th>Tanks and Temples – with ratio test</th>
<th>Tanks and Temples – without ratio test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>% Inliers</td>
<td>F-score</td>
</tr>
<tr>
<td>RANSAC</td>
<td>42.61</td>
<td>42.99</td>
</tr>
<tr>
<td>LMEADS</td>
<td>42.96</td>
<td>40.57</td>
</tr>
<tr>
<td>MLESAC</td>
<td>41.89</td>
<td>42.39</td>
</tr>
<tr>
<td>USAC</td>
<td>42.76</td>
<td>43.55</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>45.02</strong></td>
<td><strong>46.99</strong></td>
</tr>
</tbody>
</table>
KITTl

• 22 sequences of driving dataset
• 11 of them has groundtruth odometry.
Even the model trained on T&T achieves comparable performance with the state of the art networks. This demonstrates the generalization ability of the proposed method.
Dataset contains images taken from variety of cameras and from varying positions.
The data has no apparent regularity.
The data has no apparent regularity

Good performance indicates good generalization ability
• Good performance shows the generic nature of proposed algorithm – can be adapted for any geometrical modelling task based on Direct Linear Transform
SOME EXAMPLES

- Blue Dot: Outlier
- Red Dot: Inlier
- Green Line: Predicted
- Blue line: Groundtruth
What might be the reason for wrong prediction of epipolar lines?

- Failure to correctly estimate inliers/epipolar lines. Closely look at inlier at bottom.
FAILURE EXAMPLES

- Left example: Presence of noise
- Right example: Indistinct features
CRITIQUE
1. Demonstrate the advantages of integration of geometrical information into the deep learning pipeline. Also, pure data driven pipelines require huge labelled data which may not be available for a number of 3D reconstruction tasks.

2. The proposed method can learn models specific to data at hand.

3. It is robust to inlier noise distribution.

4. Good generalization ability to multiple datasets.

5. Generic nature of the proposed method encourages application in other 3D tasks which are based on Direct Linear Transform.
WEAKNESS

1. The proposed method is still dependent on the accuracy/correctness of correspondence estimator.

2. Accuracy of Fundamental Matrix estimation increases with the number of estimation modules $D$, but so does the time of evaluation.
EXTENSIONS AND FOLLOW-UPS

Deep Fundamental Matrix Estimation without Correspondences: Omid Poursaeed, Guandao Yang, Aditya Prakash, Qiuren Fang, Hanqing Jiang, Bharath Hariharan, Serge Belongie: ECCV 18
THANK YOU!!

QUESTIONS?
Fundamental Matrix Estimation

Preprocessing step $A(P)$:

$$(A(P))_i = \text{vec} \left( T \hat{p}_i (T' \hat{p}'_i)^\top \right)$$

where, $\hat{p}_i = ((p_{i1}, p_{i2}, 1)^\top$ and $\hat{p}'_i = ((p_{i3}, p_{i4}, 1)^\top$ are homogenous coordinates of correspondences $T$ and $T'$ are noramalization matrices

The model extractor $g(x)$:

$$g(x) = \arg\min_{F: \det(F)=0} \|F - T^\top (x)_{3 \times 3} T'\|_F$$

Essentially enforces rank=2 for fundamental matrix $F$. 