Comparative Analysis of Two-View and Three-View Pose Estimation Algorithms for Image-Based 3D Reconstruction

Fundamental Matrix vs Trifocal Tensor

Giovanni Versiglioni

Outline

1. The Fundamental Matrix
2. The Trifocal Tensor
3. Pose Estimation
4. Experiments

1. The Fundamental Matrix

1. The Fundamental Matrix

- Definition
- Linear computation with the Normalized Eight Point Algorithm
    1. Given at least 8 point correspondences
    2. Transform coordinates in normalized space (with normalizing matrix T)
    3. Determine normalized FM using Af = 0 system of linear equations
    4. Apply SVD to normalized FM (in order to have null determinant)
    5. Transform obtained matrix back to original space
- Optimized Gauss-Helmert Algorithm
    1. Given at least 8 point correspondences, find initial linear estimate with Normalized Eight Point Algorithm
    2. Apply Gauss-Helmert to iteratively reduce estimation error

2. The Trifocal Tensor

2. The Trifocal Tensor

- Derivation and Definition
- Trilinearities
- Linear computation with the Algebraic Minimization Algorithm
    1. Given set of point/line correspondences
    2. Compute At = 0 system of linear equations, and solve using least squares to find initial estimate
    3. Find epipoles e21 and e31 as common perpendicular to left-null vectors of the tensor's slices
    4. Build matrix E s.t. t = Ea, and find t (27-vector made up of TFT entries)
    5. Iteratively apply least squares with Levenberg-Marquardt to improve (and eventually find the optimal) epipoles pair e21 and e31 (step 3)

- Parametrizations:
    - **Ressl**
      Constraints on the correlations slices. 20 parameters, 2 constraints.
    - **Nordberg**
      Three 3x3 orthogonal matrices, transforming original tensor to sparse one with 10 non-zero parameters.
    - **Faugeras and Papadopoulo**
      12 constraints (3 third-degree, 9 sixth-degree).
    - **Ponce and Hebert**
      Three matrices, derived from intersection of three lines in space, impose constraints on the correspondence among three image points. 6 homogenous constraints.
- Optimized Gauss-Helmert Algorithm
    1. Find initial linear estimate with Algebraic Minimization Algorithm
    2. Apply Gauss-Helmert with respect to the considered parametrizations to iteratively reduce estimation error

3. Pose Estimation

3. Pose Estimation

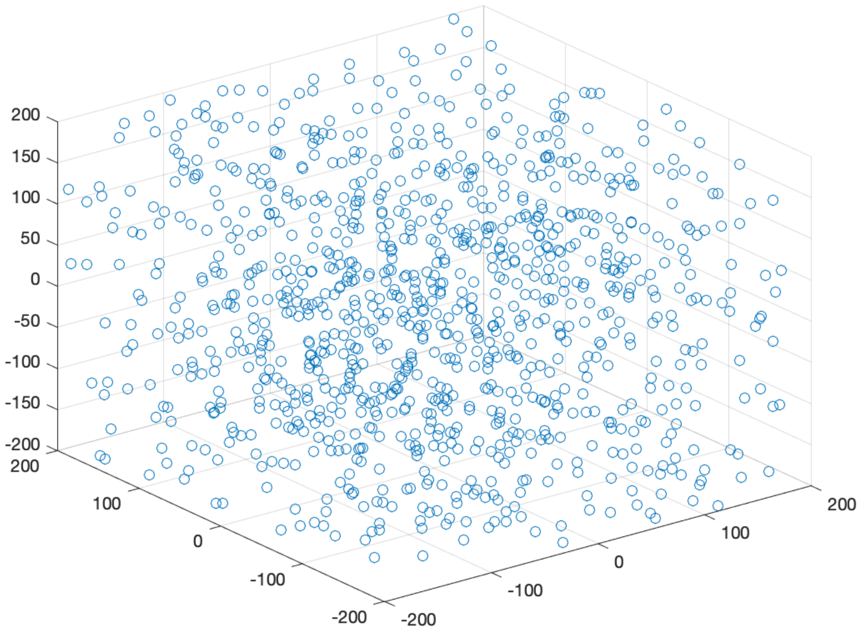
Method to extract camera poses either from the trifocal tensor or from the fundamental matrices:

1. Given a trifocal tensor, derive the epipoles (projection of first camera center onto second and third images)
2. Compute the fundamental matrices
3. Derive the essential matrices (and the calibration matrices)
4. Derive the camera orientations (rotation + translation) via SVD of the essential matrices
5. Apply Bundle Adjustment to refine the orientations, by minimizing the square reprojection error

4. Experiments

4. Experiments: Synthetic Data

- Initial setup:
  - Spatial points within a 400mm-sided cube, projected onto three views
  - Gaussian noise with 1px standard deviation
  - 50mm focal length
  - Null angle among camera centres
  - 1800x1200 pixels image, meaning 36x24 mm sensor size
- Simulations before and after Bundle Adjustment:
  - Varying Gaussian noise standard deviation
  - Varying focal length
  - Varying number of points
  - Varying angle among camera centers
- Results in Report



4. Experiments: Real Data

- Triplets of images from “fountain-P11” and “Herz-Jesu-P8” of the given EPFL Dense Multi-View Stereo Dataset
- Method Outline:
    1. Iterate through triplets, extracting point correspondences
    2. Compute metrics for each method considered (FM, TFT parametrizations)

4. Experiments: Real Data: Results (fountain-P11)

Table 1. Initial metrics with respect to the *fountain-P11* set of images.

	repr. error (px)	R error (°)	t error (°)	# iter.	time (s)
L-TFT	2.3953	0.1249	0.4048	0	0.0621
R-TFT	2.0474	0.1158	0.4003	2.8429	0.6400
N-TFT	2.1322	0.1334	0.4028	2.8000	0.6280
FP-TFT	2.3688	0.1187	0.4055	2.7714	0.6073
PH-TFT	2.0871	0.1167	0.4030	2.5857	0.5554
L-FM	1.9671	0.1149	0.3717	0	0.0273
O-FM	1.9530	0.1127	0.3658	4.9286	0.3209

Table 2. Metrics after Bundle Adjustment with respect to the *fountain-P11* set of images.

	repr. error (px)	R error (°)	t error (°)	# iter.	time (s)
L-TFT	0.2814	0.0640	0.0743	3.8143	0.0743
R-TFT	0.2814	0.0640	0.0743	3.8286	0.0720
N-TFT	0.2814	0.0640	0.0743	3.8571	0.0716
FP-TFT	0.2814	0.0640	0.0743	3.8429	0.0723
PH-TFT	0.2814	0.0640	0.0743	3.8429	0.0743
L-FM	0.2814	0.0640	0.0743	3.7714	0.0816
O-FM	0.2814	0.0640	0.0743	3.8000	0.0784

4. Experiments: Real Data: Results (Herz-Jesu-P8)

Table 3. Initial metrics with respect to the *Herz-Jesu-P8* set of images.

	repr. error (px)	R error (°)	t error (°)	# iter.	time (s)
L-TFT	4.8062	0.4589	0.8707	0	0.0506
R-TFT	3.4792	0.3966	0.6677	2.7800	0.4904
N-TFT	4.0656	0.5252	0.6917	2.6600	0.4816
FP-TFT	4.5006	0.4459	0.8324	3.4400	0.5452
PH-TFT	4.5293	0.4261	0.6682	2.3000	0.4116
L-FM	3.7624	0.4142	0.7725	0	0.0224
O-FM	3.6503	0.4196	0.7654	5.6600	0.2906

Table 4. Metrics after Bundle Adjustment with respect to the *Herz-Jesu-P8* set of images.

	repr. error (px)	R error (°)	t error (°)	# iter.	time (s)
L-TFT	0.3719	0.0635	0.0682	4.0600	0.0792
R-TFT	0.3719	0.0635	0.0682	4.0000	0.0674
N-TFT	0.3719	0.0635	0.0682	4.0400	0.0690
FP-TFT	0.3719	0.0635	0.0682	4.0600	0.0680
PH-TFT	0.3719	0.0635	0.0682	4.0000	0.0664
L-FM	0.3719	0.0635	0.0682	4.0000	0.0718
O-FM	0.3719	0.0635	0.0682	4.0200	0.0724

5. Conclusions

5. Conclusions

- Bundle Adjustment has a major influence on the performances with respect to the different mathematical structures considered
- However, without BA, TFT metrics are (in general) slightly better than FM metrics
- But, not that significantly to consider the TFT a better solutions over the FM, also considering that the FM is a much simpler and more intuitive mathematical structure