Three-view Focal Length Recovery From Homographies - Supplementary Material

Yaqing Ding[1] Viktor Kocur[2] Zuzana Berger Haladová[2] Qianliang Wu[3]
Shen Cai[4] Jian Yang[3] Zuzana Kukelova[1]

[1] Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague
{yaqing.ding, kukelzuz}@fel.cvut.cz

[2] Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava
{viktor.kocur, haladova}@fmph.uniba.sk

[3] PCA Lab, Nanjing University of Science and Technology, Nanjing, China
{wuqiangliang, csjyang}@njust.edu.cn

[4] Visual and Geometric Perception Lab, Donghua University
hammer_cai@163.com

In this supplementary material, we provide additional information promised in the main paper. More detailed information on the constraints introduced in Sec. 4 of the main paper is provided in Sec. 1. In Sec. 2 we provide the details of the proposed $\mathbf{H}_{\rho f}$ solver for Case IV. This solver was briefly introduced in Sec. 5.2 of the main paper. In Sec. 3 we provide additional synthetic experiments for Case I and II. In Sec. 4 we provide an evaluation of the proposed methods $\mathbf{H}_{\rho\rho f} + \mathbf{P}4\mathbf{P}f$ and $\mathbf{H}_{\rho f} + \mathbf{P}4\mathbf{P}f$ for Case III and IV using both synthetic and real data. Sec. 5 contains information on the dataset that we have collected and used for evaluation in Sec. 6.2 of the main paper.

1. New Constraints

Here we provide details on the constraints introduced in Sec. 4 of the main paper. We show the steps for their derivation including Macaulay2 [5] code.

To derive the constraints relating the focal lengths and the elements of the matrix $\mathbf{Q}_i$, we first create an ideal I [3] generated by the 12 polynomials extracted form the matrix equation Eq (16) in the main paper, *i.e*. the equation

$$[\mathbf{n}]_\times \mathbf{Q}_j [\mathbf{n}]_\times^\top = s_j [\mathbf{n}]_\times [\mathbf{n}]_\times^\top, \ j = 2, 3 \tag{1}$$

Note that both the left and right of (1) are symmetric matrices, hence we can get 6 equations for each j.

In the next step, the unknown elements of the normal vector $\mathbf{n}$, *i.e*., n_x, n_y, n_z, and the scale factors s_2, s_3 are eliminated from the generators of I by computing the generators of the elimination ideal $J = I \cap \mathbb{C}[q_{21}, \ldots, q_{36}]$. Here, $q_{j.}$ are the entries of $\mathbf{Q}_2, \mathbf{Q}_3$. These generators can be computed, for example, in the computer algebra software Macaulay2 [5] using the following code:

```
KK = ZZ / 30097;
R = KK[q21,q22,q23,q24,q25,q26,
q31,q32,q33,q34,q35,q36,nx,ny,nz,s2,s3]
Q2 = matrix({{q21,q22,q23},{q22,q24,q25},{q23,q25,q26}});
Q3 = matrix({{q31,q32,q33},{q32,q34,q35},{q33,q35,q36}});
Nx = matrix {{0,-nz,ny},{nz,0,-nx},{-ny,nx,0}};
eqs = flatten(Nx*Q2*transpose(Nx)-s2*Nx*transpose(Nx)
    | Nx*Q3*transpose(Nx)-s3*Nx*transpose(Nx) | nz-1) ;
I = ideal eqs;
J = eliminate(I,{nx,ny,nz,s2,s3});
```

In this case, the elimination ideal J is generated by seven polynomials g_i of degree 6 in the elements of $\mathbf{Q}_j$, $j = 2, 3$. The final constraints are only related to the 12 elements of the symmetric matrices $\mathbf{Q}_i$ (6 from $\mathbf{Q}_2$ and 6 from $\mathbf{Q}_3$).

2. Solver for Case IV

In this section, we present details on the $\mathbf{H}_{\rho f}$ solver for Case IV, which was introduced in Sec. 5.2 of the main paper.

Similar to Case III, the system of polynomials g_i, $i = 1, \ldots, 7$ can be written as $\mathbf{M}\mathbf{u} = \mathbf{0}$, where $\mathbf{M}$ is a 7×16 coefficient matrix and

$$\mathbf{u} = [1, \beta, ..., \beta^3, \alpha, \alpha\beta, ..., \alpha^3, ..., \alpha^3\beta^3]^\top, \tag{2}$$

is a vector consisting of the 16 monomials. We can choose α as the hidden variable, resulting in

$$\mathbf{A}(\alpha)\tilde{\mathbf{u}} = \mathbf{0}, \tag{3}$$

where $\mathbf{A}(\alpha)$ is a 7×4 polynomial matrix parameterized by α, and $\tilde{\mathbf{u}} = [1, \beta, ..., \beta^3]^\top$ is a vector of 4 monomials

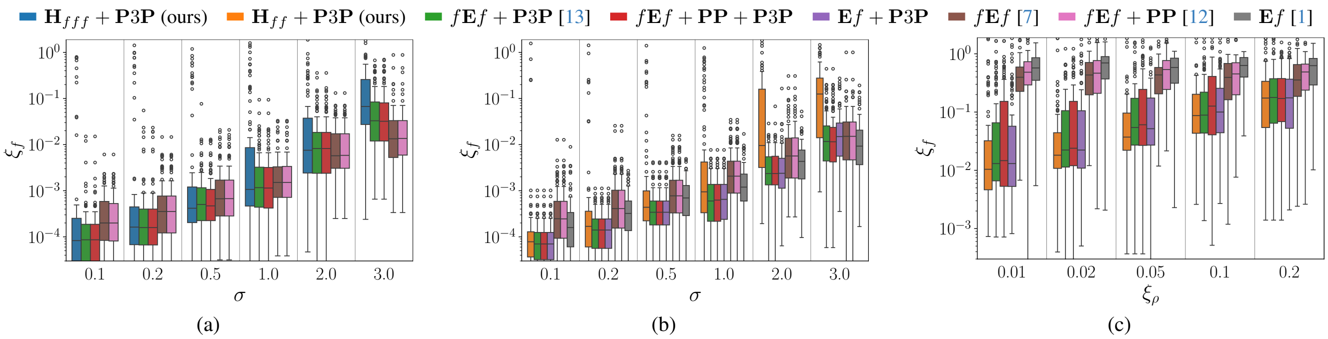

Figure 1. Focal length errors for the evaluated methods with local optimization disabled when $n_p/n = 0.5$ for varying levels of noise for Case I (a) and Case II (b). And setup for Case II with $n_p/n = 1$ with fixed noise $\sigma = 1$ as we vary the error ξ_ρ of the known focal length ρ.

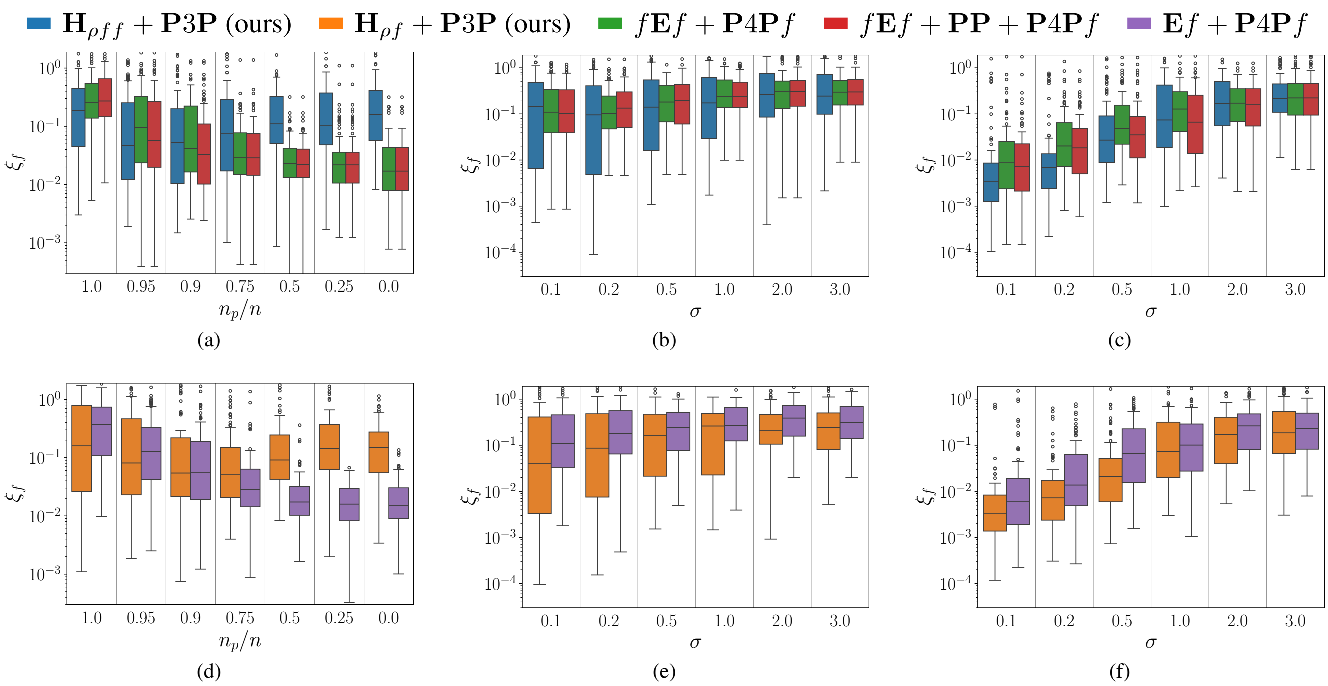

Figure 2. Focal length errors for the evaluated methods in synthetic experiments with local optimization disabled. **Case III**: (a) We vary the proportion of points which lie on the dominant plane with fixed noise $\sigma = 1$. (b, c) We vary noise σ with (b) $n_p/n = 1.0$ and (c) $n_p/n = 0.95$. **Case IV**: (d,e,c) Same setup as for Case III. The synthetic setup is described in Sec. 6.1 if the main paper.

in β without α. To solve the problem (3) as a polynomial eigenvalue problem, it is sufficient to choose four out of the seven rows in (3) to get a square matrix $\tilde{\mathbf{A}}(\alpha)$. In this case, we have

$$\tilde{\mathbf{A}}(\alpha) = \alpha^3\mathbf{A}_3 + \alpha^2\mathbf{A}_2 + \alpha\mathbf{A}_1 + \mathbf{A}_0, \tag{4}$$

where $\mathbf{A}_3, \mathbf{A}_2, \mathbf{A}_1, \mathbf{A}_0$ are 4×4 matrices. The solutions to α are given by the eigenvalues of the following 12×12 matrix

$$\begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ -\mathbf{A}_3^{-1}\mathbf{A}_0 & -\mathbf{A}_3^{-1}\mathbf{A}_1 & -\mathbf{A}_3^{-1}\mathbf{A}_2 \end{bmatrix}$$

In this way, we obtain 12 possible solutions. The remaining steps are similar to Case III. We denote this solver as $\mathbf{H}_{f\rho}$. Note that in this case, the original seven polynomial equations have only nine solutions. By selecting a subset of four polynomials, we introduced three more solutions. Still, the resulting solver is more efficient than the solver to the original seven equations, due to Gauss-Jordan elimination and computations of complex coefficients that are performed in a solver with 9 solutions.

3. Additional Synthetic Experiments for Case I and II

In Figure 1 we present additional synthetic experiments for Case I and II. We use the same setup as presented in Section 5.2 of the main paper. Experiments with smaller proportion of points on the dominant plane (Fig. 1 (a, b)) shows that some baselines perform on par with our method when sufficient off-plane points are available for estimation. Fig. 1(c) shows a comparison of the evaluated methods for Case II when we introduce error in the known focal length ρ showing that our method performs significantly better in comparison with the baselines.

4. Evaluation for Case III and Case IV

In this section we evaluate the robust estimators $\mathbf{H}_{\rho f f} + \mathbf{P}3\mathbf{P}$ for Case III and $\mathbf{H}_{f\rho} + \mathbf{P}3\mathbf{P}$ for Case IV which were presented in Sec.5.3 of the main paper. As baselines for comparison we use the combination of the $f\mathbf{E}f$solver [7] or its DEGANSAC [2] variant with the plane and parallax solver [12] together with the $\mathbf{P}4\mathbf{P}f$ solver [8] for Case III and the $\mathbf{E}f$ solver in combination with $\mathbf{P}4\mathbf{P}f$ solver [8] for Case IV.

We use the same evaluation metrics as in the main paper. Since two focal lengths are estimated jointly we use the geometric mean of their error $\xi_f = \sqrt{\xi_{f_1}\xi_{f_2}}$.

4.1. Multiple Geometrically Valid Solutions

We observe that for both Case III and Case IV a single planar scene may result in multiple geometrically valid solutions. This makes it difficult to select a single correct solution for a given set of point correspondences. This leads to generally worse performance of the proposed solvers than for Case I and Case II. Note, that this is a feature of the problems and not the solvers. Problem with recovering one focal length for Case III was mentioned also in [6].

In Case III and Case IV, in contrast to baselines that use two-view $f\mathbf{E}f$and $\mathbf{E}f$solvers and that completely fail for purely planar scenes, our solvers among the returned solutions contain the geometrically correct solution (see Figure 2 in the main paper). The proposed solvers just cannot distinguish between the returned solutions without additional information. In Sec 4.3 we show how some simple strategies using prior knowledge about the focal lengths can significantly improve performance on real-world data even for these challenging problems.

We note that this problem can also be overcome whenever the scene contains a sufficient number of off-plane points. In such cases, there is one dominant plane with some off-plane objects visible in the three views. The off-plane points can then lead to higher scores for the correct solutions and are therefore selected during RANSAC. We demonstrate this both in synthetic experiments presented in the next section and with real-world experiments using which are presented in Sec. 4.3.

4.2. Synthetic Experiments

We perform synthetic experiments with a setup similar to the one presented in Sec. 6.1 of the main paper. To better compare the performance of solvers under multiple possible valid solutions, we perform the experiment using vanilla RANSAC (without local optimization). The results of the synthetic experiments presented in Fig. 2 show that the estimators $\mathbf{H}_{\rho f f} + \mathbf{P}3\mathbf{P}$ for Case III and $\mathbf{H}_{f\rho} + \mathbf{P}3\mathbf{P}$ for Case IV perform better than the baselines when considering a planar scene as well as scenes with a dominant plane. We also note that for all solvers the accuracy of the estimated focal lengths improves as more off-plane points are added to the planar scene.

4.3. Real-World Experiments

Original | Added off-plane objects

Figure 3. We added two objects to a scene from the planar dataset and recaptured it with 8 cameras that were also used to capture the original scene for comparison.

We also perform experiments on the real-world dataset introduced in this paper. To overcome issues with multiple geometrically feasible solutions, we propose a simple strategy of acceptable field-of-view (FOV) ranges for the focal lengths. During RANSAC we simply discard all solutions with focal lengths outside of the predetermined range.

In Tab. 1 we show the results comparing our method and the baseline approaches in three different variants. As the first variant we do not discard any solutions. For the second variant, we set range of acceptable FOVs by considering the prior for focal lengths used by the popular SfM software COLMAP [11] which is set as $f_p = 1.2\max(width, height)$, which corresponds to a field of view of $\sim 45°$. To obtain a range we consider 30% increase or decrease in focal length, resulting in the acceptable field of view range from 35.5° to 61.5°. As the last variant we use a range of 50° to 70°. We chose this range since most commercially available phone cameras fall within it.

All variants were implemented in PoseLib [9]. We set the maximum epipolar threshold to 3 px. We used early

	Method		FOV Filtering	Sample	Median ξ_f	Mean ξ_f	mAA$_f$(0.1)	mAA$_f$(0.2)	Runtime (ms)
Case III	$\mathbf{H}_{f\rho\rho} + \mathbf{P3P}$	**ours**		4 triplets	0.2463	0.3579	19.34	29.68	121.62
	$f\mathbf{E}f + \mathbf{P4P}f$			6 triplets	0.2890	0.3700	16.33	25.84	**34.09**
	$f\mathbf{E}f + \mathbf{PP} + \mathbf{P4P}f$			6 triplets	0.2950	0.3740	16.04	25.46	35.66
	$\mathbf{H}_{f\rho\rho} + \mathbf{P3P}$	**ours**	35.5° - 61.5°	4 triplets	0.1478	0.2862	24.07	37.40	142.64
	$f\mathbf{E}f + \mathbf{P4P}f$			6 triplets	0.2136	0.3211	18.34	29.84	40.57
	$f\mathbf{E}f + \mathbf{PP} + \mathbf{P4P}f$			6 triplets	0.2199	0.3258	17.89	29.23	42.67
	$\mathbf{H}_{f\rho\rho} + \mathbf{P3P}$	**ours**	50° - 70°	4 triplets	**0.1171**	**0.2288**	**27.22**	**42.32**	136.19
	$f\mathbf{E}f + \mathbf{P4P}f$			6 triplets	0.1582	0.2571	22.31	35.61	37.33
	$f\mathbf{E}f + \mathbf{PP} + \mathbf{P4P}f$			6 triplets	0.1629	0.2602	21.95	35.09	39.37
Case IV	$\mathbf{H}_{f\rho} + \mathbf{P3P}$	**ours**		4 triplets	0.3286	0.4119	14.83	24.01	60.11
	$\mathbf{E}f + \mathbf{P4P}f$			6 triplets	0.3871	0.4357	11.99	19.61	**32.50**
	$\mathbf{H}_{f\rho} + \mathbf{P3P}$	**ours**	35.5° - 61.5°	4 triplets	0.1559	0.2761	20.85	35.06	98.21
	$\mathbf{E}f + \mathbf{P4P}f$			6 triplets	0.1982	0.3075	16.90	29.63	56.74
	$\mathbf{H}_{f\rho} + \mathbf{P3P}$	**ours**	50° - 70°	4 triplets	**0.1101**	**0.2257**	**26.52**	**43.30**	89.18
	$\mathbf{E}f + \mathbf{P4P}f$			6 triplets	0.1198	0.2341	24.84	41.36	50.77

Table 1. Results on the real-world dataset of planar scenes for Case III and IV. FOV filtering denotes the range of FOV values that is used to reject models within RANSAC.

	Scene	Method		Sample	Median ξ_f	Mean ξ_f	mAA$_f$(0.1)	mAA$_f$(0.2)	Runtime (ms)
Case III	Original	$\mathbf{H}_{f\rho\rho} + \mathbf{P3P}$	**ours**	4 triplets	0.3076	0.3884	15.48	24.34	133.43
		$f\mathbf{E}f + \mathbf{P4P}f$		6 triplets	**0.2494**	**0.3310**	**16.36**	**26.59**	**37.40**
		$f\mathbf{E}f + \mathbf{PP} + \mathbf{P4P}f$		6 triplets	0.2620	0.3446	15.36	24.98	39.44
	Off-plane	$\mathbf{H}_{f\rho\rho} + \mathbf{P3P}$	**ours**	4 triplets	**0.1540**	**0.3077**	**29.39**	**39.64**	142.92
		$f\mathbf{E}f + \mathbf{P4P}f$		6 triplets	0.2035	0.3127	25.08	34.84	**38.36**
		$f\mathbf{E}f + \mathbf{PP} + \mathbf{P4P}f$		6 triplets	0.2115	0.3174	24.60	34.30	40.53
Case IV	Original	$\mathbf{H}_{f\rho} + \mathbf{P3P}$	**ours**	4 triplets	**0.3843**	0.4290	**11.92**	**19.90**	69.22
		$\mathbf{E}f + \mathbf{P4P}f$		6 triplets	0.4011	**0.4283**	11.00	18.98	**32.95**
	Off-plane	$\mathbf{H}_{f\rho} + \mathbf{P3P}$	**ours**	4 triplets	**0.2381**	**0.3557**	**21.81**	**31.71**	69.97
		$\mathbf{E}f + \mathbf{P4P}f$		6 triplets	0.3150	0.3933	17.30	25.64	**35.54**

Table 2. Evaluation results on the original Book scene from the planar scenes dataset and its modified version with objects added in order to introduce off-plane scenes (see Fig. 3). The modified version of the scene was captured by eight cameras. Therefore, for evaluation of the original scene we only consider triplets of images taken by the same eight cameras.

termination with 0.9999 confidence and a minimum of 100 iterations and a maximum of 1000.

For all of the proposed variants and both cases our methods $\mathbf{H}_{\rho f f} + \mathbf{P3P}$ and $\mathbf{H}_{\rho f} + \mathbf{P3P}$ show superior performance in terms of the accuracy of the estimated focal lengths compared to the baselines. We also note that the performance of all evaluated methods significantly improves when filtering solutions based on the predetermined field of view ranges. Showcasing how a simple strategy can significantly improve the accuracy of all evaluated methods.

Evaluation using a scene with off-plane points

We also perform additional evaluation with one scene which is similar to the Book scene from our planar dataset, but we include additional objects (see Fig. 3) to introduce some off-plane points. This sequence was captured by 8 of the cameras used to capture the planar dataset. It contains 87 images in total and we used them to generate 5066 triplets for Case III and 1473 triplets for Case IV.[1] We perform a comparison between the results obtained on this scene with off-plane objects and results obtained on the original planar scene. For the original planar scene we only consider triplets using the same cameras as in the scene with off-plane objects. Tab. 2 shows a comparison of the results for all evaluated methods. The results show a significant improvement of all methods when off-plane objects are introduced in the scene. This shows that the problem becomes easier when the scene is not fully planar, but retains a significant dominant plane with some off-plane points. In this scenario, our methods show significantly better performance over the baselines.

5. Real World Dataset

This section provides the detailed information about the dataset used for real-world evaluation presented in Sec. 6.2 of the main paper. The descriptions of the cameras and

[1] Note that this scene is not included in the dataset presented in Sec. 5 due to its different nature, *i.e.* containing additional non-planar objects.

					Images							Triplets		
ID	Description	FOV	Width	Height	Asphalt	Boats	Book	Facade	Floor	Papers	Calib	Case I	Case II/III	Case IV
IPhoneOldBack	Apple IPhone SE (2nd generation) back camera	62.8°	4032	3024	20	19	22	20	20	20	14	3000	2806	2822
IPhoneOldFront	Apple IPhone SE (2nd generation) front camera	56.5°	3088	2320	20	18	18	23	17	21	18	2755	2984	3089
IPhoneZBHBack	Apple IPhone SE (3rd generation) back camera	63.3°	4032	3024	19	19	21	20	20	20	20	3000	2761	2781
IPhoneZBHfront	Apple IPhone SE (3rd generation) front camera	56.2°	3088	2320	20	20	22	19	21	20	20	2389	2900	3065
LenovoTabletBack	Tablet Lenovo TB-X505F back camera	59.7°	2592	1944	20	20	21	18	20	17	13	3000	2968	3079
LenovoTabletFront	Tablet Lenovo TB-X505F front camera	62.3°	1600	1200	✗	21	21	✗	16	✗	20	1500	1803	2189
MotoBack	Motorola Moto E4 Plus back camera	64.7°	4160	3120	20	23	22	21	22	20	37	2013	2529	2903
MotoFront	Motorola Moto E4 Plus front camera	71.0°	2592	1952	✗	18	19	✗	✗	✗	19	1000	1400	1732
Olympus	Olympus uD600,S600 compact digital camera	49.1°	2816	2112	19	23	24	17	21	21	23	3000	3145	3171
SamsungBack	Samsung Galaxy S5 Mini back camera	56.1°	3264	1836	✗	20	28	✗	20	20	18	2000	2356	2594
SamsungFront	Samsung Galaxy S5 Mini front camera	69.4°	1920	1080	19	24	19	23	19	20	20	2798	3023	3120
SamsungGlossyBack	Samsung Galaxy S III Mini back camera	53.8°	2560	1920	20	21	19	20	19	20	21	3000	3073	3180
SamsungGlossyFront	Samsung Galaxy S III Mini front camera	55.6°	640	480	21	20	26	20	20	21	20	3000	1790	1726
DellWide	Dell Precision 7650 notebook camera	80.0°	1280	720	✗	21	22	✗	22	✗	20	1500	1254	1446
SonyTelescopic	Sony α5000 digital camera with 55-210mm Lens	23.5°	5456	3064	20	20	20	22	18	✗	20	1517	1664	1815
		Total			218	307	324	223	275	220	303	35472	18219	12876
		Triplets Case I			3574	7500	7500	5500	5898	5500	✗			
		Triplets Case II			649	5100	4856	2538	2555	2521	✗			
		Triplets Case IV			252	4256	3871	1472	1618	1407	✗			

Table 3. Summary of our evaluation dataset. The table shows the number of included images per scene per camera and the number of extracted triplets. The last three columns indicate how many triplets for a given case contain an image from a given camera (*e.g.* for Case IV we use 1815 triplets for which at least one of the images was taken using Dell Precision 7650 notebook camera). The fourth row from bottom denotes the total number of images per scene in the dataset and and in the last three columns the total number of triplets per case. The last three rows show how many triplets are included for each scene.

Asphalt	Boats	Book	Facade	Floor	Papers	DinoBook
99.9%	93.1%	96.8%	98.0%	95.4%	96.3%	74.3%

Table 4. Share of planar points for images in individual scenes.

dataset statistics regarding the total number of images and extracted triplets are provided in Tab. 3. The dataset contains 1870 images of 4 indoor and 2 outdoor planar scenes captured with 14 calibrated cameras. We purposefully select some scenes to be more challenging (*e.g.* repeating patterns in Floor, few significant landmarks in Asphalt). In total we use provide 66 567 image triplets for evaluation of the different cases.

In Table **??**, we provide the proportion of points on a plane for each scene. The proportions were determined using RealityCapture SfM software to reconstruct the scene using images from all cameras. We selected the point cloud generated for the largest component, found the dominant plane and counted the number of points within a manually determined distance from the plane. As seen in the statistics, for all scenes in the main paper the proportion of points is $> 93\%$ with the DinoBook scene containing significantly lower ratio.

5.1. Calibration

To calibrate the cameras, we used a standard checkerboard pattern printed on hard plastic. We manually removed blurry or otherwise unsuitable calibration images from the dataset. We calibrated the cameras using [14]. During calibration, we used the assumption of square pixels (*i.e.* $f_x = f_y$). We also modeled tangential and radial distortion to obtain more accurate focal lengths. All used cameras exhibited low distortion so we use the original distorted images for evaluation to better reflect accuracy in real scenarios where cameras are expected to have low and unknown distortion. The images used for calibration, as well as the calibration code and estimated intrinsics, will be made available with the dataset.

5.2. Triplet Point Correspondence Extraction

To obtain triplet correspondences we used SuperPoint [4] with inference in the original image resolution keeping at most 2048 best keypoints. We matched the keypoints using LightGlue [10] to first perform pairwise matches. Since SuperPoint was trained to only be rotationally invariant up to $45°$ rotations we have extracted pairwise matches by rotating one of the images four times with a step of $90°$ and selecting the orientation which produced the largest number of matches. Afterwards, we kept only those correspondences which were matched across all three pairs thus producing triplets. We will provide the extracted correspondences as part of the dataset upon release.

References

[1] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. 3d reconstruction from image collections with a single known focal length. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1803–1810. IEEE, 2009. 2

[2] Ondrej Chum, Tomas Werner, and Jiri Matas. Two-view geometry estimation unaffected by a dominant plane. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 772–779. IEEE, 2005. 3

[3] David A Cox, John Little, and Donal O'shea. *Using algebraic geometry*. Springer Science & Business Media, 2005. 1

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabi-

novich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 5

[5] Daniel R. Grayson and Michael E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at http://www2.macaulay2.com. 1

[6] Janne Heikkilä. Using sparse elimination for solving minimal problems in computer vision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 76–84, 2017. 3

[7] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Polynomial eigenvalue solutions to minimal problems in computer vision. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2012. 2, 3

[8] Zuzana Kukelova, Jan Heller, and Andrew Fitzgibbon. Efficient intersection of three quadrics and applications in computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1799–1808, 2016. 3

[9] Viktor Larsson and contributors. PoseLib - Minimal Solvers for Camera Pose Estimation, 2020. 3

[10] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023. 5

[11] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 3

[12] Akihiko Torii, Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. The six point algorithm revisited. In *Computer Vision–ACCV 2010 Workshops: ACCV 2010 International Workshops, Queenstown, New Zealand, November 8-9, 2010, Revised Selected Papers, Part II 10*, pages 184–193. Springer, 2011. 2, 3

[13] Charalambos Tzamos, Daniel Barath, Torsten Sattler, and Zuzana Kukelova. Relative pose of three calibrated and partially calibrated cameras from four points using virtual correspondences. *arXiv preprint arXiv:2303.16078*, 2023. 2

[14] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000. 5