

Partially calibrated semi-generalized pose from hybrid point correspondences

Supplementary Material

Snehal Bhayani¹ Torsten Sattler² Viktor Larsson³

Janne Heikkilä¹ Zuzana Kukelova⁴

¹Center for Machine Vision and Signal Analysis, University of Oulu, Finland

²Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague

³Computer Vision and Geometry Group, Department of Computer Science, ETH Zürich

⁴Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague

This supplementary material provides additional details for the solvers presented in the main paper. Sec. 1 describes a special reparameterization of the product \mathbf{KR} of the calibration matrix \mathbf{K} and the rotation matrix \mathbf{R} , which was used to derive the $\mathbf{H32}f$ solver in the main paper. Sec. 2 derives the $\mathbf{H51}f$ solver for the [1], . . . , [4] camera configurations (the [5] camera configuration was studied in the main paper). Sec. 3 provides additional details about the synthetic experiments from the main paper. In this supplementary material, we follow the notations and conventions used in the main paper.

1. Reparameterization of \mathbf{KR}

Here, we follow the method proposed in [7] to reparameterize \mathbf{KR} . For the sake of brevity, we write $\mathbf{R}_K = \mathbf{KR}$. We can split the rotation matrix as $\mathbf{R} = \mathbf{R}_\theta \mathbf{R}_\rho$, where \mathbf{R}_θ represents the rotation around the z -axis by an angle θ and \mathbf{R}_ρ denotes the rotation around an axis in the $x - y$ plane. Thus, we have $\mathbf{R}_K = \mathbf{KR}_\theta \mathbf{R}_\rho$. We can express \mathbf{KR}_θ as

$$\begin{aligned} \mathbf{KR}_\theta &= \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} f \cos \theta & -f \sin \theta & 0 \\ f \sin \theta & f \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (1)$$

Introducing two new variables r_1 and r_2 , we write $r_1 = f \cos \theta$ and $r_2 = f \sin \theta$. We have thus reparameterized the angle θ and the focal length f using r_1 and r_2 . The rotation matrix \mathbf{R}_ρ in the $x - y$ plane can be parameterized using the quaternion [1, r_3 , r_4 , 0]. Together, we thus have a reparameterization of $\mathbf{R}_K = \mathbf{KR}$ using four variables r_1, r_2, r_3 and r_4 .

Apart from this, we also tried other formulations for the rotation matrix, such as the Cayley parameterization, the axis-angle representation, Euler angles, *etc.* But the reparameterization described here was the most helpful in generating smaller solvers (smaller elimination template size in the case of Gröbner basis-based methods [5, 6] or Generalized Eigenvalue Problem (GEP) size in the case of the resultant-based method [1]) by removing symmetry and halving the number of solutions.

rameterization described here was the most helpful in generating smaller solvers (smaller elimination template size in the case of Gröbner basis-based methods [5, 6] or Generalized Eigenvalue Problem (GEP) size in the case of the resultant-based method [1]) by removing symmetry and halving the number of solutions.

2. $\mathbf{H51}f$

In the main paper, we considered the $\mathbf{H51}f$ case for the camera configuration where all five 2D-2D point correspondences are detected by the same pinhole camera \mathcal{G}_i , *i.e.*, the [5] camera configuration. Here, we derive solvers for the remaining camera configurations, *i.e.*, [1], . . . , [4]. Note that in [3], only the most general configuration [1] for this case was discussed. Here, we individually studied the special configurations, [2], [3] and [4], which allowed us to derive specific solvers with fewer solutions and smaller solver sizes. The sizes of the proposed solvers for these four camera configurations for the $\mathbf{H51}f$ case are provided in the main paper.

The constraints induced by a 2D-2D point correspondence $\mathbf{p}_j \leftrightarrow (\mathbf{q}_{ij}, \mathbf{t}_{G_i})$ and a 2D-3D point correspondence $\mathbf{p}_l \leftrightarrow \mathbf{X}_l^G$ are given by

$$\mathbf{R}(\beta_{ij} \mathbf{q}_{ij} + \mathbf{t}_{G_i}) + \mathbf{t} = \alpha_j \mathbf{K}^{-1} \mathbf{p}_j, \quad (2)$$

$$\mathbf{R} \mathbf{X}_l^G + \mathbf{t} = \alpha_l \mathbf{K}^{-1} \mathbf{p}_l. \quad (3)$$

These correspond to Eqs. (4) and (5) in the main paper. Eliminating the depths β_{ij} and α_j from Eq. (2), and the depth α_l from Eq. (3), we have

$$(\mathbf{p}_j)^\top [\mathbf{KR} \mathbf{q}_{ij}]_\times (\mathbf{KR} \mathbf{t}_{G_i} + \mathbf{Kt}) = 0, \quad (4)$$

$$[\mathbf{p}_l]_\times (\mathbf{KR} \mathbf{X}_l^G + \mathbf{Kt}) = \mathbf{0}. \quad (5)$$

For brevity, let us write $\mathbf{t}_K = \mathbf{Kt}$ and $\mathbf{R}_K = \mathbf{KR}$. Then we

have

$$\mathbf{p}_j^\top [\mathbf{R}_K \mathbf{q}_{ij}]_\times (\mathbf{R}_K \mathbf{t}_{G_i} + \mathbf{t}_K) = 0, \quad (6)$$

$$\mathbf{p}_l^\top [\mathbf{R}_K \mathbf{X}_l^G]_\times \mathbf{t}_K = \mathbf{0}. \quad (7)$$

Eq. (6) gives us one equation and Eq. (7) gives us two linearly independent equations in \mathbf{t}_K and \mathbf{R}_K . Therefore, from five 2D-2D point correspondences $\mathbf{p}_j \leftrightarrow (\mathbf{q}_{ij}, \mathbf{t}_{G_i})$ for $j = 1, \dots, 5$ and one 2D-3D point correspondence $\mathbf{p}_l \leftrightarrow \mathbf{X}_l^G$, $l = 6$, we have a total of seven equations in seven unknowns, *i.e.*, the elements of the vector \mathbf{t}_K and the four-variable reparameterization of \mathbf{R}_K .

As mentioned in the main paper, we solved this problem by trying three different parameterizations, *i.e.*, *Rotation & translation*, *Homography* and *Fundamental matrix*. We attempted the *Rotation & translation* parameterization, which corresponds to Eqs. (2) and (3), for all camera configurations. We attempted the *Homography* parameterization for the camera configurations [2], [3] and [4], and the *Fundamental matrix* parameterization for the camera configuration [4].

First, we briefly describe the *Homography* and *Fundamental matrix* parameterizations. For both parameterizations, we translated the coordinate system of the generalized camera \mathcal{G} such that the center of the camera \mathcal{G}_i that detected the largest number of 2D-2D point correspondences coincides with the origin, *i.e.*, $\mathbf{t}_{G_i} = [0 \ 0 \ 0]^\top$.

Homography parameterization: Among the 2D-2D correspondences detected by the same camera \mathcal{G}_i , we select three 2D-2D correspondences without loss of generality (w.l.o.g.) and consider the scene plane defined by these three correspondences. This plane allows us to define a homography \mathbf{H} between \mathcal{G}_i and the pinhole camera \mathcal{P} . Each of these three 2D-2D correspondences gives us two constraints on the homography matrix \mathbf{H} , *i.e.*, two linearly independent equations that are linear in the elements of \mathbf{H} . These six equations allow us to parameterize the homography matrix using a three-dimensional null space. Our approach here is similar to that of our proposed **H13f** solver in Sec. 3.1 in the main paper. We generate the polynomial ideal in the entries of \mathbf{R} , \mathbf{t} , and \mathbf{N} as unknown variables. After eliminating \mathbf{R} , we derive an elimination ideal in $\mathbf{H}_K = \mathbf{K}\mathbf{H}$, \mathbf{t} , and \mathbf{N} . The three-variable parameterization of the homography matrix is then substituted in the generators of this elimination ideal and also in the constraints induced by the remaining 2D-2D and the 2D-3D point correspondence. The resulting set of equations define our minimal solver.

Fundamental matrix parameterization: We use the constraints induced by the five 2D-2D point correspon-

dences¹ detected by the same camera \mathcal{G}_i , allowing us to parameterize the fundamental matrix using a four-dimensional null space. The fundamental matrix is expressed as

$$\mathbf{F} = \lambda \mathbf{K}^{-1} [\mathbf{t}]_\times \mathbf{R}, \quad (8)$$

where $\lambda \in \mathbb{R}$ is an unknown variable. Let the first two columns of \mathbf{F} be \mathbf{f}_1 and \mathbf{f}_2 . We can express the translation vector \mathbf{t} and the rotation matrix \mathbf{R} as

$$\mathbf{t} = \lambda_1 \mathbf{K}^{-1} [\mathbf{f}_1]_\times \mathbf{f}_2, \quad (9)$$

$$\mathbf{R} = \lambda_2 \mathbf{K}^{-1} (\mathbf{K} \mathbf{t} \times \mathbf{F} + \mathbf{K} \mathbf{t} \mathbf{v}^\top), \quad (10)$$

where \mathbf{v} is an unknown 3×1 vector and $\lambda_1, \lambda_2 \in \mathbb{R}$ are unknown scalars. We substitute these expressions for \mathbf{t} and \mathbf{R} in the constraints induced by the remaining 2D-2D point correspondence as well as the second constraint induced by the 2D-3D point correspondence. These two equations, along with the determinant constraint on the fundamental matrix, $\det \mathbf{F} = 0$, and the trace constraint on the corresponding essential matrix, $2\mathbf{E}\mathbf{E}^\top \mathbf{E} - \text{trace}(\mathbf{E}\mathbf{E}^\top)\mathbf{E} = 0$, define the initial polynomial system representing our minimal solver.

However both these parameterizations lead to large solver templates and the polynomial systems have more than four variables. In comparison, the **Rotation & translation** parameterization leads to the smallest polynomial systems in terms of variables and polynomial degree, and thus the smallest solver size. Out of all the parameterizations we tested, we thus describe our solver based on the **Rotation & translation** parameterization in most detail. In this solver the first step is to eliminate the vector \mathbf{t} from the constraints, defined by Eqs. (2) and (3).

2.1. [1] camera configuration

Let us first consider the most generic camera configuration [1], where each 2D-2D point is detected by a different pinhole camera \mathcal{G}_i , with $i = 1, \dots, 5$. Thus, we have $j = 1, \dots, 5$ and $i = 1, \dots, 5$.

Using the constraint imposed by the 2D-3D point correspondence (Eq. (3)), we express $\mathbf{t}_K = \alpha_6 \mathbf{p}_6 - \mathbf{R}_K \mathbf{X}_6^G$ and substitute it in the constraints induced by the 2D-2D point correspondences. We have five constraints, each of which is of the same form as Eq. (6). We thus have five equations in the unknowns α_6, r_1, r_2, r_3 , and r_4 . Let us express them in matrix form as

$$\mathbf{M}_6 \begin{bmatrix} \alpha_6 \\ 1 \end{bmatrix} = \mathbf{0}_{5 \times 2}, \quad (11)$$

where the matrix \mathbf{M}_6 is a 5×2 matrix whose entries are polynomials in the unknowns r_1, r_2, r_3 , and r_4 . We compute the

¹Note, that the fifth 2D-2D point correspondence is obtained by projecting the 3D point to the camera \mathcal{G}_i .

determinants of each submatrix of size 2×2 from M_6 , resulting in a set of ten polynomials in r_1, r_2, r_3 , and r_4 . Let us denote this set as E_6 .

Similarly, we obtain $\mathbf{t}_k = \alpha_j \mathbf{p}_j - \beta_{ij} \mathbf{R}_k \mathbf{q}_{ij} - \mathbf{R}_k \mathbf{t}_{G_i}$ from the constraint of the form of Eq. (2), induced by the 2D-2D point correspondence $\mathbf{p}_j \leftrightarrow (\mathbf{q}_{ij}, \mathbf{t}_{G_i})$. We substitute this form of \mathbf{t}_k in the four equations of the form of Eq. (6) for the remaining four 2D-2D point correspondences and in the two equations of the form of Eq. (7) for the 2D-3D point correspondence. We thus have six equations in unknown variables $\alpha_j, \beta_{ij}, r_1, r_2, r_3$, and r_4 . We can write them in matrix form as

$$M_j \begin{bmatrix} \alpha_j \\ \beta_{ij} \\ 1 \end{bmatrix} = \mathbf{0}_{6 \times 3}, \quad (12)$$

where the matrix M_j is a 6×3 matrix whose entries are polynomials in the unknown variables r_1, r_2, r_3 , and r_4 . The set of determinants of all 3×3 submatrices gives us the set E_j of 20 polynomials in r_1, r_2, r_3 , and r_4 .

We perform the above step for $j = 1, \dots, 5$. In each case, we have matrices M_j which lead to the set of polynomials, E_j , each containing 20 polynomials. The union of all these sets, $E = \bigcup_{j=1}^6 E_j$, denotes the polynomial system for our proposed solver. It consists of 110 polynomials in four variables r_1, r_2, r_3 , and r_4 . However, the ideal generated by E is non-zero dimensional. We next propose an approach to introduce extra polynomials in order to saturate the ideal.

Ideal saturation : Let the columns of the matrix M_6 in Eq. (11) be \mathbf{m}_1 and \mathbf{m}_2 . We have observed that the last column \mathbf{m}_2 vanishes if $r_3 = 0$ & $r_4 = 0$. Therefore, \mathbf{m}_2 can be expressed as $\mathbf{m}_2 = r_3 \mathbf{m}'_2 + r_4 \mathbf{m}''_2$. We have

$$M_6 \begin{bmatrix} \alpha_6 \\ 1 \end{bmatrix} = \mathbf{0} \implies [\mathbf{m}_1 \quad \mathbf{m}'_2 \quad \mathbf{m}''_2] \begin{bmatrix} \alpha_6 \\ r_3 \\ r_4 \end{bmatrix} = \mathbf{0}. \quad (13)$$

Let $N_6 = [\mathbf{m}_1 \quad \mathbf{m}'_2 \quad \mathbf{m}''_2]$. The determinants of all possible 3×3 submatrices of N_6 lead to a set of ten polynomials in r_1, r_2, r_3 , and r_4 which do not vanish for $r_3 = 0$ & $r_4 = 0$. We augment E with these ten polynomials. By abuse of notation, let us denote the augmented set as E . Note that this is a rather complicated system, with polynomial degrees from 7 up to 11 in r_1, r_2, r_3 , and r_4 .

Gauss-Jordan elimination: Let us order these polynomials in E in an ascending order of their degree. Let B be the set of monomials in E . We can construct its vector form \mathbf{b} by ordering the monomials by their degree in descending order. Thus, we can rewrite E in matrix form as $X \mathbf{b}$, where the rows of X are indexed by the polynomials and their columns are indexed by the entries in \mathbf{b} . We

perform Gauss-Jordan elimination on X , resulting in 45 non-zero rows. These rows correspond to 45 linearly independent polynomials, which define our simplified system. It consists of polynomials with degrees ranging from 5 up to 11. We generated solvers for this set of 45 polynomials and also for its subsets. In this way, we observed that the smallest solver can be generated by using polynomials of degrees only up to 10. Using Macaulay 2 [2], we verified that this system has up to 56 solutions².

Using the Gröbner basis-based automatic generator [5, 6] we obtained a minimal solver from this polynomial system whose elimination template is of size 506×562 (see Tab. 1 in the main paper). Using the resultant-based method [1], we obtained a solver whose GEP is of size 537×537 .

2.2. [2], [3], [4] camera configurations

Our approach for deriving the solvers for the other three camera configurations, [2], [3] and [4], is similar to the case of the [1] configuration. For the [2] camera configuration, our approach remains exactly the same, but the polynomial system has 50 solutions (see Tab. 1 in the main paper)). But for the [3] camera configuration, each set E_j , $j = 1, \dots, 3$, has one extra polynomial. This is because each matrix M_j , $j = 1, \dots, 3$, in Eq. (12) has the form

$$M_j = \begin{bmatrix} * & * & 0 \\ * & * & 0 \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}. \quad (14)$$

We compute the determinant of the 2×2 submatrix in the top-left corner and add it to E_j . This results in 113 polynomials in E instead of 110. The subsequent steps remain the same, *i.e.*, Gauss-Jordan elimination and selecting a subset of the polynomials for generating the minimal solver. This polynomial system has up to 50 solutions.

In the same way, for the [4] configuration, each matrix M_j , $j = 1, \dots, 4$, in Eq. (12) has the form

$$M_j = \begin{bmatrix} * & * & 0 \\ * & * & 0 \\ * & * & 0 \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}. \quad (15)$$

In this case, the determinant of the first 3×3 submatrix is zero, but we have determinants of the three 2×2 submatrices in the top-left corner which we add to E_j . This results in

²In contrast, this problem was reported to have 112 solutions in [3]. Thanks to our reparameterization of the matrix \mathbf{R}_k in Sec. 1, we were able to halve the number of solutions. This means that the solver has to perform Eigendecomposition of a matrix of size 56 instead of 112.

118 polynomials in E instead of 110. The subsequent steps are the same, *i.e.*, Gauss-Jordan elimination and selecting a subset of the polynomials for generating the minimal solver. This polynomial system has up to 38 solutions. The sizes of the proposed solvers for these camera configurations for the $\mathbf{H51f}$ case are provided in the main paper.

3. Synthetic experiments

We provide additional details for the synthetic experiments presented in the main paper. In the main paper, we showed the errors in the estimated rotations for three different motions, *i.e.*, random motion, forward motion, and sideways motion, in the presence of increasing 3D point noise. For the same experiments, **Row 1** of Fig. 1 and **Row 2** of Fig. 1 respectively show the errors in the estimated focal length f and translation \mathbf{t} . For all three motions, in the presence of increasing 3D point noise, the $\mathbf{H51f}[5]$ solver and the proposed $\mathbf{H32f}$ and $\mathbf{H13f}$ solvers are much more stable than the SOTA $\mathbf{P4Pf}$ absolute pose solver, with the $\mathbf{H51f}[5]$ solver slightly outperforming the $\mathbf{H32f}$ and $\mathbf{H13f}$ solvers.

Additionally, we also provide details for the synthetic experiment where we varied the image px noise while fixing the 3D point noise that was mentioned in the paper. We set the camera motion to be random for this experiment. We compared the stability of the proposed $\mathbf{H32f}$, $\mathbf{H13f}$, and $\mathbf{H51f}[5]$ solvers against the SOTA absolute pose solver $\mathbf{P4Pf}$ [4]. We have used exactly the same scene setup as in the main paper: We tested two scenarios, by fixing the Gaussian noise in the 3D points at 0.1% and 0.5% of the scene depth, and for each scenario, varied the amount of px Gaussian noise in the measured 2D points. In Fig. 1, we show the errors in the translation vector \mathbf{t} , the rotation matrix \mathbf{R} , and the focal length f in **Row 3** and **Row 4** for Gaussian noise in 3D points at respectively 0.1% and 0.5% of the scene depth. As expected, the performance of our hybrid solvers, which are using also 2D-2D matches, is decreasing with increasing image noise. Nevertheless, even for small levels of noise in 3D point positions (0.1% and 0.5% of the scene depth), which favors the $\mathbf{P4Pf}$ solver, our hybrid solvers still perform competitively compared to $\mathbf{P4Pf}$.

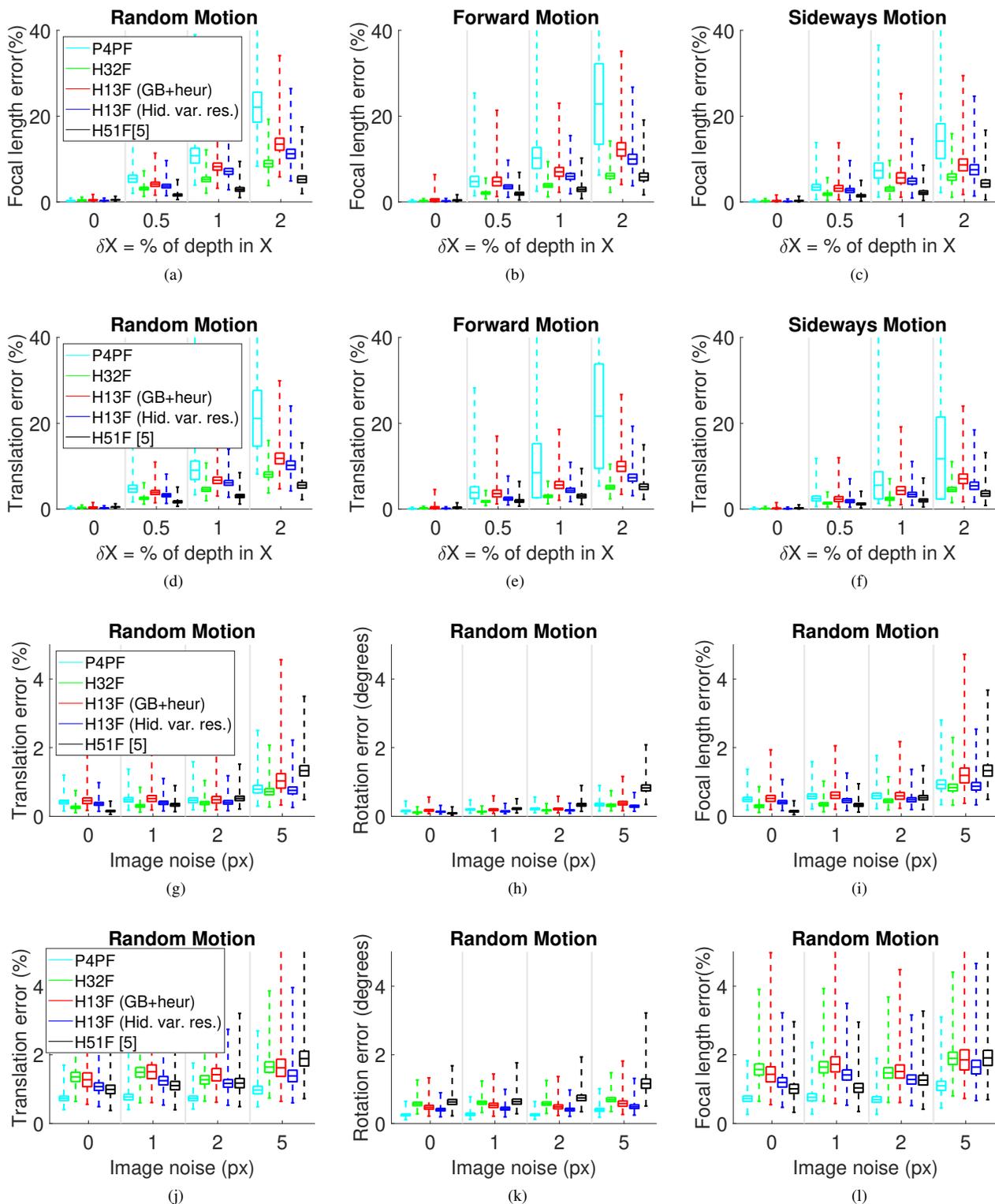


Figure 1. Error in focal length (**Row 1**) and translation (**Row 2**) in the presence of increasing 3D point noise for random motion (**a**), forward motion (**b**), and sideways motion (**c**). Error in translation (**d,g**), rotation (**e,h**), and focal length (**f,i**) in the presence of increasing 2D px noise for noise in 3D points set to be 0.1% (**Row 3**) and 0.5% (**Row 4**) of the scene depth.

References

- [1] Snehal Bhayani, Zuzana Kukelova, and Janne Heikkilä. Computing stable resultant-based minimal solvers by hiding a variable. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6104–6111, 2021.
- [2] Daniel R. Grayson and Michael E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at <http://www.math.uiuc.edu/Macaulay2/>.
- [3] Klas Josephson, Martin Byrod, Fredrik Kahl, and Kalle Astrom. Image-based localization using hybrid feature correspondences. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [4] Zuzana Kukelova, Jan Heller, and Andrew Fitzgibbon. Efficient intersection of three quadrics and applications in computer vision. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [5] Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *CVPR*, volume 2, page 4, 2017.
- [6] Viktor Larsson, Magnus Oskarsson, Kalle Åström, Alge Wallis, Zuzana Kukelova, and Tomas Pajdla. Beyond gröbner bases: Basis selection for minimal solvers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3945–3954, 2018.
- [7] Enliang Zheng and Changchang Wu. Structure From Motion Using Structure-Less Resection. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.