Uncertainty-Aware Camera Pose Estimation from Points and Lines: Supplementary Materials

Alexander Vakhitov[1] Luis Ferraz Colomina[2] Antonio Agudo[3] Francesc Moreno-Noguer[3]
[1]SLAMCore Ltd., UK
[2]Kognia Sports Intelligence, Spain
[3]Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Spain

1. Introduction

In the following supplementary materials, we describes results of the additional experiments, including more data on a real experiment for points and lines described in the main paper, and some additional baseline comparisons in a synthetic setup; also, we give theoretical details on the methods. Please also consider the **MATLAB code** for the synthetic experiments, and a **video**, shortly described next.

2. Video description

The video show the first 200 frames of the KITTI 01 sequence. We run RANSAC, then inlier filtering, then a solver (either a standard DLS or a proposed DLSU), and *standard* pose refinement. We plot the detections considered as inlier ones by RANSAC, as well as the projections of the features considered as inliers by the methods. Comparing the initial inlier detections and the projections of the inliers after the inlier filtering, one can see that the proposed method has better alignment between the projected and the detected features. For some frames, it selects the feature sets which are more diverse in terms of point depth, often choosing the features which were not present in the original inlier set. See Table 3 for some examples.

3. Details of Methods

In this part, we will describe additional theoretical details behind the proposed methods.

3.1. EPnPU

In the following section, we outline the uncertainty-aware PCA procedure we use in this method. Recall, that we use an isotropic approximation to the point uncertainty $\mathbf{\Sigma}_{\mathbf{X}_i} = \sigma_i^2 \mathtt{I}$.

1. **The covariance-weighted mean point**. As long as the mean point is the point with minimal sum of distances to the points, we modify this definition to use point covariances:

$$\bar{\mathbf{x}} = \text{argmin}_{\mathbf{X}} \sum_{i=1}^{n_{pt}} \sigma_i^{-2} \|\mathbf{x}_i - \mathbf{x}\|^2, \tag{1}$$

$$\bar{\mathbf{x}} = \frac{1}{\sum_{i=1}^{n_{pt}} \sigma_i^{-2}} \sum_{j=1}^{n_p t} \sigma_j^{-2} \mathbf{x}_i, \tag{2}$$

and the covariance $\sigma_{\bar{\mathbf{x}}}^2 \mathtt{I}$ of $\bar{\mathbf{x}}$ is

$$\sigma_{\bar{\mathbf{x}}}^2 = \frac{1}{\sum_{i=1}^{n_{pt}} \sigma_i^{-2}}. \tag{3}$$

2. Compute the covariances $\sigma_{\bar{\mathbf{x}}_i}^2 \mathtt{I}$ for the centered points $\bar{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{x}$. According to the definition,

$$\sigma_{\bar{\mathbf{x}}_i}^2 = \text{cov} \left(\mathbf{x}_i - \frac{1}{\sum_{i=1}^{n_{pt}} \sigma_i^{-2}} \sum_{j=1}^{n_p t} \sigma_j^{-2} \mathbf{x}_i \right), \tag{4}$$

and transforming this, we get

$$\sigma_{\bar{\mathbf{x}}_i}^2 = 1 + \sigma_i^2 - \frac{2}{\sum_{i=1}^{n_{pt}} \sigma_i^{-2}}. \tag{5}$$

3. The j^{th} principal direction is a solution to the following covariance-weighted problem:

$$\mathbf{z}_j = \text{argmax} \sum_{i=1}^{n_{pt}} \sigma_{\bar{\mathbf{x}}_i}^{-2} \left(\bar{\mathbf{x}}_i^T \mathbf{z}_j \right)^2, \tag{6}$$

subject to $\mathbf{z}_j^T \mathbf{z}_i = 0, \ i = 1, \ldots, j-1; \|\mathbf{z}_j\| = 1$, which follows from computing the covariance of the residuals $\text{cov}\left(\bar{\mathbf{x}}_i^T \mathbf{z}_j\right) = \sigma_{\bar{\mathbf{x}}_i}^{-2}$, as explained in the paper.

Next, we move to describing the details of the computations for the DLSU method.

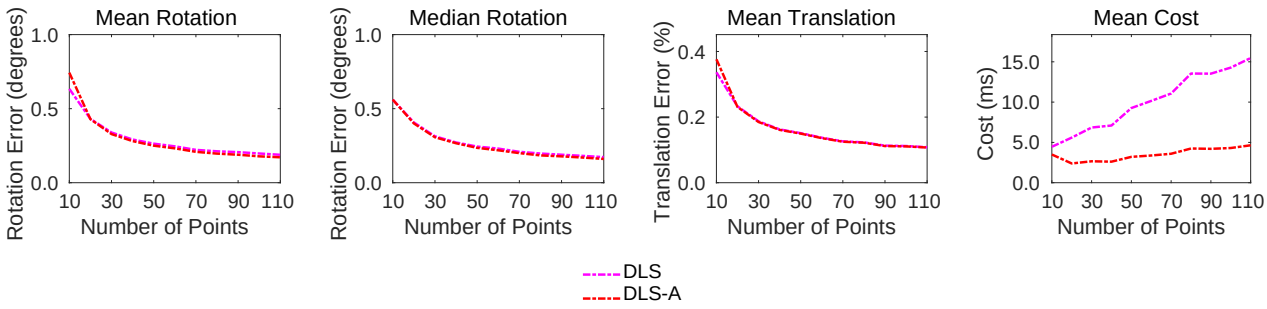


Figure 1. Pose errors and running times in a synthetic experiment with 2D noise, same conditions as in the main paper, Figure 3, top row. We compare the original DLS and the algebraic DLS-A based on algebraic distance. The latter is much faster, and the former gives slight benefits in mean errors for small point counts.

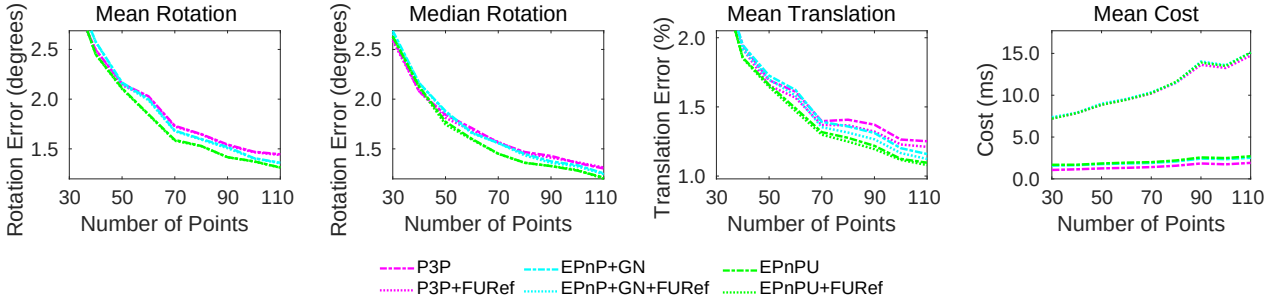


Figure 2. Pose errors and running times in a synthetic experiment with 2D+3D noise, same conditions as in the main paper, Figure 3, central row. We compare the *uncertain* and the *full uncertain* refinement methods (+FURef) for the pipelines based on P3P, EPnP+GN and proposed EPnPU solvers. *Full uncertain* refinement has highly similar accuracy to the *uncertain* refinement.

3.2. DLSU

In the following text, we give the detailed step-by-step formulas for the DLSU method. After formulating the cost as given in the main paper, we set the gradient of the cost by $\mathbf{t}$ equal $\mathbf{0}$ and express the translation through the rotation parameters

$$\mathbf{t} = -\mathtt{T}^{-1}\mathtt{A}\mathrm{vec}(\mathtt{R}(\mathbf{s})), \tag{7}$$

where $\mathtt{T} = \sum_{i=1}^{n_k} \mathtt{T}_k^T \mathbf{\Sigma}_{\mathbf{r}_k}^{-1} \mathtt{T}_k$, $\mathtt{A} = \sum_{i=1}^{n_k} \mathtt{T}_k^T \mathbf{\Sigma}_{\mathbf{r}_k}^{-1} \mathtt{A}_k$.

The gradient of the cost by the rotation rotation parameters is set to be equal zero as well

$$\mathtt{A}_k^T \mathbf{\Sigma}_{\mathbf{r}_k}^{-1} (\mathtt{A}_k - \mathtt{T}_k \mathtt{T}^{-1} \mathtt{A}) \mathrm{vec}(\mathtt{R}(\mathbf{s})) \nabla_{\mathbf{s}} \mathrm{vec}(\mathtt{R}(\mathbf{s})) = \mathbf{0}. \tag{8}$$

As long as the equations are homogeneous with respect to the vectorized rotation $\mathrm{vec}(\mathtt{R}(\mathbf{s}))$, we multiply them by $1 + \|\mathbf{s}\|^2$ following the original DLS approach. We get a 3^{rd}-order polynomial system with three unknown components of $\mathbf{s}$. We solve it with a generated Groebner solver, and compute $\mathbf{t}$ using (7).

3.3. Covariance-aware line triangulation

While line representation with the 3D endpoints is clearly non-minimal, because the domain of lines in 3D is 4-dimensional, but two 3D endpoints together give a dimension of 6, the endpoint-based parameterization is still used in practice.

We assume that we are given the camera poses $\mathtt{R}_i, \mathbf{t}_i$, $i = 1, \ldots, N_l$, and the line segments detected in the corresponding images, defined by the pairs $(\mathbf{x}_i^s, \mathbf{x}_i^e)$ of the endpoints in the image plane. We propose to constrain the 3D endpoints to project to the detected segment endpoints on the first image. For the other images, the projections of the endpoints should belong to the detected lines, not necessarily projecting to the 2D endpoints. This way, the endpoints would encode the spatial location of the detected segment better. We use the following cost for line triangulation, which is a sum of 2D covariance-weighted point-based residuals for the first camera, and line-based residuals for the other cameras: $L_{ln}(\mathbf{P}, \mathbf{Q}) =$

$$\|\bar{\mathbf{r}}_{\mathbf{x}_1^s}^{pt}(\mathbf{p})\|_{\mathbf{\Sigma}_{\mathbf{x}_1^s}}^2 + \|\bar{\mathbf{r}}_{\mathbf{x}_1^e}^{pt}(\mathbf{q})\|_{\mathbf{\Sigma}_{\mathbf{x}_1^e}}^2 + \sum_{i=2}^{N_l} \|\bar{\mathbf{r}}_i^{ln}(\mathbf{p}, \mathbf{q})\|_{\mathbf{\Sigma}_{\mathbf{l},i}}^2, \tag{9}$$

where we denote the point projection residuals $\mathbf{r}_{\mathbf{x}_1^s}^{pt}(\mathbf{p}) = \mathbf{r}_{\mathbf{x}_1^s}^{pt}(\mathbf{p}, \mathtt{R}_1, \mathbf{t}_1)$ and $\mathbf{r}_{\mathbf{x}_1^e}^{pt}(\mathbf{q}) = \mathbf{r}_{\mathbf{x}_1^e}^{1}(\mathbf{q}, \mathtt{R}_1, \mathbf{t}_1)$, as given in (19), main paper, and $\bar{\mathbf{r}}_i^{ln}(\mathbf{p}, \mathbf{q}) = \bar{\mathbf{r}}_i^{ln}(\mathbf{p}, \mathbf{q}, \mathtt{R}_i, \mathbf{t}_i)$, as given in (2), main paper.

We find $\mathbf{p}, \mathbf{q}$ using Levenberg-Marquardt-based optimization of $L_{ln}(\mathbf{p}, \mathbf{q})$, initializing with the result of the DLT-based line triangulation as explained in [3].

For the error propagation, we follow a general scheme,

	Points								Points + 2D Uncertainty				Points + Full Uncertainty, Proposed							
	P3P [5]		EPnP [6]		DLS [4]		OPnP [10]		CEPPnP [1]		MLPnP [8]		EPnPU*		EPnPU		DLSU*		DLSU	
	e_{rot}	e_{trans}	e_{rot}	e_{trans}	e_{rot}	e_{trans}	e_{rot}	e_{trans}	e_{rot}	e_{trans}	e_{rot}	e_{trans}	e_{rot}	e_{trans}	e_{rot}	e_{trans}	e_{rot}	e_{trans}	e_{rot}	e_{trans}
KITTI [2], sequences 00-02																				
N	3.6	17.3	1.4	9.8	1.3	7.1	1.3	7.1	3.1	17.9	**1.0**	**4.7**	1.2	8.3	1.7	7.9	1.3	7.9	1.5	6.2
S	0.9	4.7	0.9	4.3	0.9	4.2	0.9	4.2	1.0	5.2	0.9	4.2	0.9	4.2	0.9	4.3	**0.9**	4.2	0.9	***4.1***
U	0.9	4.7	0.8	4.5	0.8	4.4	0.8	4.4	0.9	5.0	0.8	4.5	0.8	4.5	0.8	4.5	***0.8***	4.4	0.8	**4.4**
TUM [7], 'freiburg1' sequences																				
N	13.3	2.7	9.1	1.2	8.9	1.1	8.9	1.1	9.1	1.2	8.7	1.0	8.9	1.0	8.9	1.1	8.8	1.0	**8.7**	**1.0**
S	8.6	1.0	8.6	0.9	8.6	0.9	8.6	0.9	8.6	0.9	8.6	0.9	8.6	0.9	8.5	0.9	**8.5**	0.9	8.5	**0.9**
U	8.6	0.9	8.5	0.9	8.6	0.9	8.6	0.9	8.6	0.9	8.6	0.9	8.5	0.9	8.5	0.9	8.5	0.9	***8.5***	***0.9***

Table 1. Motion estimation from 2D-3D point correspondences on KITTI [2] TUM [7] in terms of median absolute rotation e_{rot} (in $0.1\times$deg.) and translation e_{trans} (in cm.) errors. We compare proposed full uncertainty-aware methods against point-based PnP and 2D uncertainty-aware methods in isolation (**N**), with standard (**S**) and proposed uncertain (**U**) refinement. Methods with '*' receive a pose from RANSAC, best for the dataset is in bold italic, best for each protocol (N,S or U) is in bold. The new methods outperform the baselines in most metrics.

e.g. [3], Chapter 5, getting

$$\mathbf{\Sigma}_{\mathbf{p},\mathbf{q}} = (\mathtt{J}^T(\mathbf{p},\mathbf{q})\mathtt{J}(\mathbf{p},\mathbf{q}))^{-1}, \tag{10}$$

where $\mathtt{J}(\mathbf{p},\mathbf{q})$ denotes the Jacobian of the inverse covariance-weighted residuals. We obtain $\mathbf{\Sigma}_{\mathbf{p}}$ as a left-upper 3×3 block of $\mathbf{\Sigma}_{\mathbf{p},\mathbf{q}}$, and $\mathbf{\Sigma}_{\mathbf{q}}$ as the right-lower 3×3 block of the same matrix, which is an approximation indeed, motivated in the main paper by the simplicity of the formulation and the efficiency of computations.

4. Additional experiments

In this section, we give additional experimental results.

4.1. Median errors for points

In Table 1 we present the median errors for the real experiment on KITTI and TUM described in the main paper. While the proposed methods mostly outperform the competitive methods, the gap in terms of median errors is smaller compared to the gap in mean errors. While MLPnP excels in isolation on KITTI, it has much higher runtime because it runs reprojection cost refinement inside, while other solvers do not.

4.2. Full results on lines

Due to limited space in the main paper, we present here the results for the points + lines pipeline on TUM and KITTI datasets, same sequences as for the points in the main paper, see Table 2. The proposed solvers outperform the baselines in translation errors on both datasets. On TUM, the media rotation errors are similar for all tested methods, while on KITTI the proposed solvers have better rotation accuracy.

4.3. DLS Modifications

We compare the original object space error-based DLS solver [4] and our modification, DLS-A, which uses the algebraic error. We get DLS-A from the DLSU solver by providing it with unit matrices as residual covariances. See Figure 1 for the results. The proposed DLS-A is 2-3 times faster depending on a point number, but also is slightly inferior for the low number of points with respect to the original DLS method.

4.4. Refinement modifications

In this section, we compare the *full uncertain* and the proposed *uncertain* refinement methods, as described in Section 3.5 of the main paper. *Full uncertain* refinement is implemented using finite-difference approximation of the Jacobian and based on MATLAB lsqnonlin function, implementing a Levenberg-Marquardt method, while *uncertain* refinement is implemented as Gauss-Newtwon iterations, with additional iterative re-computation of the residual covariances. The experiment is run following the setting of the main paper (2D noise + 3D noise, central row of the Figure 3, main paper). We compare a pipelines with P3P, EPnP+GN and proposed EPnPU solvers. The results in Figure 2 suggest, that there are no major differences in accuracy of the methods.

References

[1] L. Ferraz, X. Binefa, and F. Moreno-Noguer. Leveraging feature uncertainty in the PnP problem. In *Proceedings of the BMVC 2014 British Machine Vision Conference*, pages 1–13, 2014. 3

[2] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 3, 4

[3] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2, 3

[4] J. A. Hesch and S. I. Roumeliotis. A direct least-squares (DLS) method for PnP. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 383–390. IEEE, 2011. 3

		Points+Lines				Points+Lines+Uncertainty			
		EPnPL [9]		OPnPL [9]		DLSLU*		EPnPLU*	
		e_{rot}	e_{trans}	e_{rot}	e_{trans}	e_{rot}	e_{trans}	e_{rot}	e_{trans}
KITTI [2], sequences 00-02									
mean	N	**2.65**	40.26	9.54	727.26	5.88	**17.34**	2.73	20.25
	S	1.74	12.77	7.93	344.78	5.19	13.86	***1.73***	***9.71***
med.	N	1.52	15.75	**1.46**	7.98	1.49	**6.41**	1.71	8.37
	S	0.89	4.87	0.86	4.20	***0.84***	***4.04***	0.85	4.21
TUM [7], sequences 'freiburg1'									
mean	N	11.73	1.85	10.74	1.52	**9.84**	**1.24**	12.27	1.59
	S	11.13	1.38	10.43	1.29	***9.77***	***1.18***	11.88	1.39
med.	N	9.06	1.26	8.93	1.12	**8.73**	**0.93**	8.87	1.06
	S	8.64	0.92	***8.58***	0.92	***8.58***	0.92	***8.58***	***0.91***

Table 2. Motion estimation from 2D-3D point and line correspondences on KITTI [2] sequences 00-02 and TUM [7], 'freiburg1' sequences. We report the rotation errors in $0.1\times$degrees and translation errors in cm, for the solvers in isolation (N) and after *standard* refinement (S). The proposed uncertainty-aware solvers outperform the uncertainty-free baselines OPnPL and EPnPL in most metrics on KITTI and in mean metrics on TUM. Median rotation errors on TUM are similar, but the proposed solvers benefit from lower median translation errors.

[5] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2969–2976. IEEE, 2011. 3

[6] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81(2):155–166, 2009. 3

[7] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012. 3, 4

[8] S. Urban, J. Leitloff, and S. Hinz. Mlpnp - a real-time maximum likelihood solution to the perspective-n-point problem. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3:131–138, 2016. 3

[9] A. Vakhitov, J. Funke, and F. Moreno-Noguer. Accurate and linear time pose estimation from points and lines. In *European Conference on Computer Vision*, pages 583–599. Springer, 2016. 4

[10] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi. Revisiting the PnP problem: a fast, general and optimal solution. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2344–2351. IEEE, 2013. 3

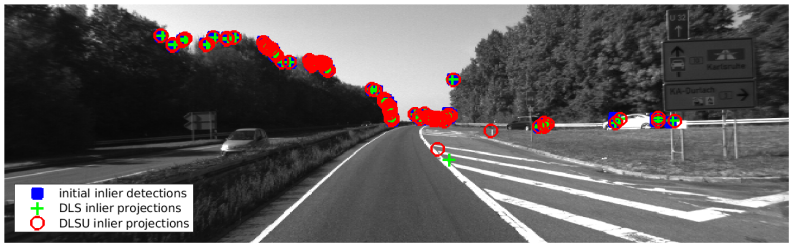
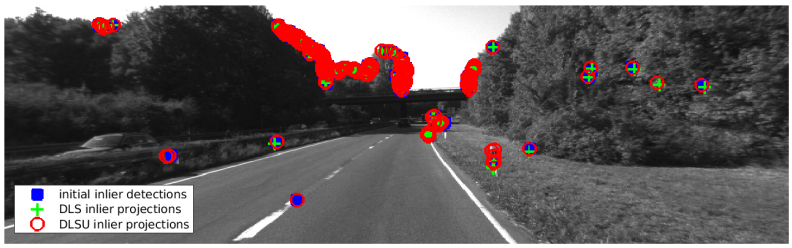
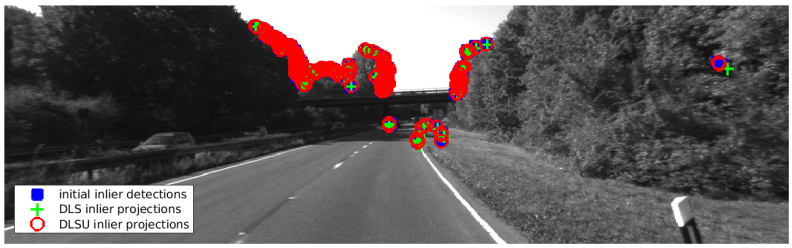
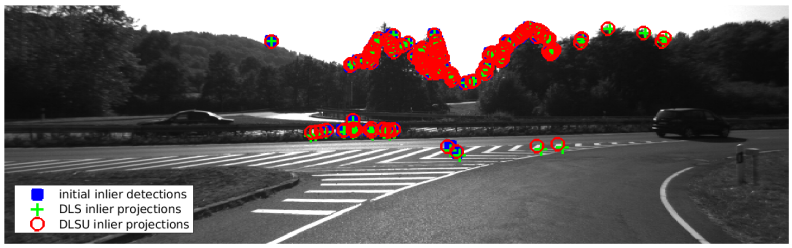
Image	Δe_{trans}, cm.
	28
	78
	14.1
	4.0
	-2.4

Table 3. We compare the DLS and DLSU filtered inlier sets reprojected onto the images, and also plot the initially estimated inlier detections. In the right column, see the improvement of absolute error by DLSU as compared to DLS, after the standard refinement. The inlier projections of DLSU are more aligned with the detections; DLSU selects closer features more often.