Generalized Pose-and-Scale Estimation using 4-Point Congruence Constraints: Supplementary Material

Victor Fragoso
Microsoft
victor.fragoso@microsoft.com

Sudipta N. Sinha
Microsoft
sudipta.sinha@microsoft.com

In this supplementary document, we discuss details of the coplanar case polynomial solver (see Sec. 1) and describe the implementation details for the general polynomial solver (see Sec. 2).

1. Specialized Method for Coplanar Points

Recall, that we are given four rays, parameterized using four 3D points, $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, denoting the projection centers and four 3D unit vectors, $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$, $\mathbf{u}_4$, denoting the ray directions. We are also given four coplanar 3D points, $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_4$ from which we get ratios r_1, r_2, K_{1213} and K_{1234} (see paper for the details). We need to compute four unknown scalar values, s_1, s_2, s_3, s_4, such that

$$\mathbf{y}_1 = \mathbf{p}_1 + s_1\mathbf{u}_1 \tag{1}$$
$$\mathbf{y}_2 = \mathbf{p}_2 + s_2\mathbf{u}_2 \tag{2}$$
$$\mathbf{y}_3 = \mathbf{p}_3 + s_3\mathbf{u}_3 \tag{3}$$
$$\mathbf{y}_4 = \mathbf{p}_4 + s_4\mathbf{u}_4 \tag{4}$$

and each $\mathbf{y}_i = T\mathbf{x}_i$ for $i = 1 \ldots 4$ where T is an unknown 3D similarity transformation.

Now, recall that in the main paper, for the coplanar points case, we presented three linear constraints and one quadratic constraint in s_1, s_2, s_3, s_4. These equations are repeated here for convenience (see Equations 5 and 6 respectively):

$$\begin{aligned}(1-r_1)(\mathbf{p}_1 + s_1\mathbf{u}_1) + r_1(\mathbf{p}_2 + s_2\mathbf{u}_2) &= \\ (1-r_2)(\mathbf{p}_3 + s_3\mathbf{u}_3) + r_2(\mathbf{p}_4 + s_4\mathbf{u}_4).\end{aligned} \tag{5}$$

The quadratic equation is

$$(\boldsymbol{\beta}_{12} - K_{1213}\boldsymbol{\beta}_{13})^\intercal \mathbf{s} = 0, \tag{6}$$

where $\mathbf{s} = \left[s_1^2\ s_2^2\ s_3^2\ s_4^2\ s_1s_2\ s_1s_3\ s_1s_4\ s_2s_3\ s_2s_4\ s_3s_4\ s_1\ s_2\ s_3\ s_4\ 1\right]^\intercal$.

In order to derive our specialied method for coplanar points, we treated s_1, s_2, s_3 as variables and rewrote the three linear equations in Equation 5 in matrix form as follows:

$$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}. \tag{7}$$

In Equation 7, the terms d_1, d_2 and d_3 are linear functions of s_4. We then consider closed form solutions of the linear system in Equation 7 based on Cramer's Rule. Thus, we obtain closed form expressions for s_1, s_2 and s_3 in terms of s_4, which take the following form:

$$s_1 = G_1s_4 + H_1 \tag{8}$$
$$s_2 = G_2s_4 + H_2 \tag{9}$$
$$s_3 = G_3s_4 + H_3. \tag{10}$$

Although, we have omitted the expressions for the six terms G_1, H_1, G_2, H_2, G_3 and H_3 here, they can be derived using basic algebraic manipulation and then plugging in all the values into the formulae for Cramer's rule. The next step is to substitute the expressions of s_1, s_2 and s_3 from Equation 8, 9 and 10 into the quadratic equation in Equation 6. This gives us a new quadratic equation in s_4:

$$As_4^2 + Bs_4 + C = 0 \tag{11}$$

where, the coefficients A, B and C are defined as follows:

$$A = C_1G_1^2 + C_2G_2^2 + C_3G_3^2 + C_4G_1G_2 + C_5G_1G_3 \tag{12}$$

$$\begin{aligned}B =\ & 2\left(C_1G_1H_1 + C_2G_2H_2 + C_3G_3H_3\right) \\ & + C_4(G_1H_2 + G_2H_1) + C_5(G_1H_3 + G_3H_1) \\ & + C_6G_1 + C_7G_2 + C_8G_3\end{aligned} \tag{13}$$

$$\begin{aligned}C =\ & C_1H_1^2 + C_2H_2^2 + C_3H_3^2 + C_4H_1H_2 + C_5H_1H_3 \\ & + C_6H_1 + C_7H_2 + C_8H_3 + C_9.\end{aligned} \tag{14}$$

In the above expressions for A, B and C, the terms C_k for $k = 1 \dots 9$ are defined as follows:

$$C_1 = 1 - K_{1213} \tag{15}$$
$$C_2 = 1 \tag{16}$$
$$C_3 = -K_{1213} \tag{17}$$
$$C_4 = -2 \cdot \mathbf{u}_1^\intercal \mathbf{u}_2 \tag{18}$$
$$C_5 = 2 \cdot K_{1213} \cdot \mathbf{u}_1^\intercal \mathbf{u}_3 \tag{19}$$
$$C_6 = 2 \cdot \mathbf{u}_1^\intercal \mathbf{p}_6 - 2 \cdot K_{1213} \cdot \mathbf{u}_1^\intercal \mathbf{p}_8 \tag{20}$$
$$C_7 = -2 \cdot \mathbf{u}_2^\intercal \mathbf{p}_6 \tag{21}$$
$$C_8 = 2 \cdot K_{1213} \cdot \mathbf{u}_3^\intercal \mathbf{p}_8 \tag{22}$$
$$C_9 = (\mathbf{p}_6^\intercal \mathbf{p}_6) - K_{1213} \cdot (\mathbf{p}_8^\intercal \mathbf{p}_8) \tag{23}$$

where $\mathbf{p}_6 = \mathbf{p}_1 - \mathbf{p}_2$ and $\mathbf{p}_8 = \mathbf{p}_1 - \mathbf{p}_3$. The next step in our method is to find the roots of the quadratic equation shown in Equation 11. We then check for positive values for s_4 and then substitute them into Equations 8, 9 and 10 to get values of s_1, s_2 and s_3. We return solutions where these three values are also positive.

2. Polynomial Solver

As explained in Section 3 in the main paper, we need to solve a polynomial system consisting of four quadratic polynomials in $\mathbf{s}$. To obtain a solver for this polynomial system, we used autogen, the automatic Gröbner-based polynomial solver generator from Larsson *et al*. [1]. While autogen can generate problem instances with random integer coefficients, we found that such instances were not representative of the geometry underlying our polynomial system and the generated solver produced inaccurate results. To address this issue, we generated our own synthetic problem instances using the following steps.

The polynomial system stated in Equations (16) and (17) in the main paper can be arranged in matrix form, $A \cdot \mathbf{s} = 0$, where the matrix $A \in \mathbb{R}^{4\times 15}$ holds the coefficients of the polynomial system. We generated synthetic problem instances, following the protocol described in the main paper. Given K problem instances and their respective coefficient matrices $A_k, \forall k = 1, \dots, K$, we computed the average coefficient matrix $A_{\text{avg}} = \frac{1}{l}\sum_{i=1}^{K} A_k$ and then multiplied every element of the matrix A_{avg} by a scale factor S before rounding the nonzero entries to the nearest integers. We set K and S to the following values: $K = 100$ and $S = 50$.

We observed that the value of the random coefficients of the polynomial system generated in this way, appeared to be representative of the underlying problem. All nonzero entries in the resulting problem instance were integers and the autogen package produced a stable and accurate solver from it. The obtained polynomial solver uses an elimination template matrix with 97 rows and 113 columns and a 16×16 action matrix. The polynomial solver returns up to 16 solutions and we only keep those that are real and positive.

References

[1] V. Larsson, K. Astrom, and M. Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 820–829, 2017. 2