Supplementary Material

1 Derive the problem formulation

Our derivations are completed through Matlab. First, we define all the constant terms and variables in the QPEP [1] and calculate their residuals.

```
syms r1 r2 r3 real
syms b1 b2 b3 real
syms n1 n2 n3 real
syms q0 q1 q2 q3 real
syms t1 t2 t3 lambda real
syms b1 b2 real
syms r1 r2 r3 real
syms fx fy cx cy real
syms A11 A12 A13 A14 real
syms A21 A22 A23 A24 real
syms A31 A32 A33 A34 real
syms B11 B12 B13 B14 real
syms B21 B22 B23 B24 real
syms B31 B32 B33 B34 real

r = [r1; r2; r3];
b = [b1; b2; b3];
nv = [n1; n2; n3];
q = [q0; q1; q2; q3];
tt = [t1; t2; t3];
RR = q2R(q);
X = [
RR, tt;
zeros(1, 3), 1];
image_pt = [b1, b2];
world_pt = [r1, r2, r3];
K = [fx, 0, 0;
0, fy, 0;
cx, cy, 1];
A = [
A11, A12, A13, A14;
A21, A22, A23, A24;
A31, A32, A33, A34;
0, 0, 0, 1;
];
B = [
B11, B12, B13, B14;
B21, B22, B23, B24;
B31, B32, B33, B34;
0, 0, 0, 1;
```

```
];
......
costs=[J1(X,A,B), J2(image_pt, world_pt, K, RR, tt, scale), J3(q, tt, r.', b.', nv.'),
    ....]
```

Finally, the coefficients of the rotation and translation terms are extracted as follows:

```
[Coefficients, Terms] = coeffs(sum(costs), [q0;q1;q2;q3; t1;t2;t3])
```

The terms in this equation correspond to the coefficients in the paper's loss function: $\mathcal{L}(\boldsymbol{q}, \boldsymbol{t}) = \boldsymbol{V}_{35}^{\top}\boldsymbol{q}^4 + \boldsymbol{V}_{40}^{\top}\mathrm{vec}(\boldsymbol{q}^2 \otimes \tilde{\boldsymbol{t}}) + \boldsymbol{V}_{6}^{\top}\boldsymbol{t}^2 + \boldsymbol{V}_{3}^{\top}\boldsymbol{t} + C$ Similarly, $\mathcal{L}s1$ and $\mathcal{L}s2$ can be obtained using the same approach.

2 Additional derivation examples showcasing the proposed method

We introduce Lagrange multipliers to $\mathcal{L}_{s1}$ to obtain equation (1) in our paper, and then take the partial derivatives of this equation with respect to the rotation and translation terms, respectively:

```
Jacob = jacobian(J, [q0;q1;q2;q3; t1;t2;t3]);
[Coefficients1, Terms1] = coeffs(Jacob(1) , [q0;q1;q2;q3; t1;t2;t3])
[Coefficients2, Terms2] = coeffs(Jacob(2) , [q0;q1;q2;q3; t1;t2;t3])
...
[Coefficients6, Terms6] = coeffs(Jacob(6) , [q0;q1;q2;q3; t1;t2;t3])
[Coefficients7, Terms7] = coeffs(Jacob(7), [q0;q1;q2;q3; t1;t2;t3])
```

Extracting Terms 1-4 and Terms 5-7 from the equation corresponds to equations (2) and (3) in our paper, respectively.

By extracting Terms 5-7, we obtain a system of three equations with three unknowns, which allows us to express the translation components in terms of q0, q1, q2, and q3.

```
[coeft1, mont1] = coeffs(Jacob(5), [tt;scale]);
g1 = coeft1(1 : 4).';
[coeftq1, montq1] = coeffs(expand(Jacob(5) - g1.' * [tt;scale] ), q);
[coeft2, mont2] = coeffs(Jacob(6), [tt;scale]);
g2 = coeft2(1 : 4).';
[coeftq2, montq2] = coeffs(expand(Jacob(6) - g2.' * [tt;scale] ), q);
[coeft3, mont3] = coeffs(Jacob(7), [tt;scale]);
g3 = coeft3(1 : 4).';
[coeftq3, montq3] = coeffs(expand(Jacob(7) - g3.' * [tt;scale] ), q);
[coeft4, mont4] = coeffs(Jacob(9), [tt;scale]);
g4 = coeft4(1 : 4).';
[coeftq4, montq4] = coeffs(expand(Jacob(9) - g4.' * [tt;scale] ), q);
G = - [g1.'; g2.'; g3.';g4.'];
pinvG = sym('pinvG', [4, 4]);
showSyms(symvar(pinvG));
coefs_tq = sym('coefs_tq', [4, 10]);
showSyms(symvar(coefs_tq));
ts = pinvG * coefs_tq * montq1.';
t1 = ts(1);
t2 = ts(2);
t3 = ts(3);
scale = ts(4);
```

By substituting the expressions for the translation components back into the corresponding parts of equation (2), we can derive the form of equation (4) in our paper.

```
Jacob = eval(Jacob);
[coef_Jacob1_qt, mon_Jacob1_qt] = coeffs(Jacob(1) + lambda * q0, [q; tts]);
[coef_Jacob2_qt, mon_Jacob2_qt] = coeffs(Jacob(2) + lambda * q1, [q; tts]);
[coef_Jacob3_qt, mon_Jacob3_qt] = coeffs(Jacob(3) + lambda * q2, [q; tts]);
[coef_Jacob4_qt, mon_Jacob4_qt] = coeffs(Jacob(4) + lambda * q3, [q; tts]);
```

3 Proof the Lemma

Given the description of the theorem, our objective is to demonstrate that for a polynomial function $f(x_1, x_2, \ldots, x_m) = C\prod_{i=1}^{m} x_i^{p_i}$, the product of its n^{th} order partial derivatives with respect to all variables and $vec(S^{\otimes n})$ is equal to $n!$ times the function itself. Here, C is a constant, p_i are non-negative integers, and $\sum_{i=1}^{m} p_i = n$.

To establish this, it is essential to consider all possible n^{th} order partial derivatives. For each specific set of $(k_1, k_2, \ldots, k_m)$ that satisfies $\sum_{i=1}^{m} k_i = n$, the n^{th} order mixed partial derivative of f (provided that all $k_i \leq p_i$) is given by:

$$\frac{\partial^n f}{\partial x_1^{k_1} \partial x_2^{k_2} \ldots \partial x_m^{k_m}} = C \cdot \prod_{i=1}^{m} \frac{p_i!}{(p_i - k_i)!} \cdot x_i^{p_i - k_i},$$

This is due to the fact that for each variable x_i, if we take the k_i^{th} partial derivative, the remaining power is $p_i - k_i$, accompanied by a coefficient $\frac{p_i!}{(p_i - k_i)!}$. Should $k_i > p_i$, the result is zero, in accordance with the definition of partial derivatives.

Next, we turn our attention to $vec(S^{\otimes n})$. This vector comprises all elements of $S^{\otimes n}$, where S is the set formed by the variables of f. Consequently, each element within $vec(S^{\otimes n})$ corresponds to the coefficient of a specific n^{th} order partial derivative, represented by the multinomial coefficient $\binom{n}{k_1, k_2, \ldots, k_m}$. This coefficient is the number of ways to distribute k_i derivatives to each x_i from the n derivatives. Here, the multinomial coefficient $\binom{n}{k_1, k_2, \ldots, k_m}$ is defined as $\frac{n!}{k_1! k_2! \ldots k_m!}$.

Now, we compute the $\mathbf{D}^{(n)} \cdot vec(S^{\otimes n})$:

$$\mathbf{D}^{(n)} \cdot vec(S^{\otimes n}) = \sum_{\sum_{i=1}^{m} k_i = n} \left(C \cdot \prod_{i=1}^{m} \frac{p_i!}{(p_i - k_i)!} \cdot x_i^{p_i - k_i} \right) \cdot \left(\frac{n!}{k_1! k_2! \ldots k_m!} \right).$$

Because we have a complete permutation of $n!$, it will be multiplied by each $\frac{p_i!}{(p_i - k_i)!}$. However, to compute the dot product, we must also consider the corresponding $x_i^{k_i}$ terms in $vec(S^{\otimes n})$. It is noted that the product of these terms yields the powers of the original polynomial f. This is due to each term $x_i^{p_i - k_i} \cdot x_i^{k_i}$ ultimately resulting in $x_i^{p_i}$. Consequently, we obtain:

$$\mathbf{D}^{(n)} \cdot vec(S^{\otimes n}) = C \cdot n! \cdot \prod_{i=1}^{m} x_i^{p_i} = n! \cdot f.eee$$

References

[1] Jin Wu, Yu Zheng, Zhi Gao, Yi Jiang, Xiangcheng Hu, Yilong Zhu, Jianhao Jiao, and Ming Liu, *Quadratic pose estimation problems: Globally optimal solutions, solvability/observability analysis, and uncertainty description*, IEEE Transactions on Robotics, vol. 38, no. 5, pp. 3314–3335, 2022.