Homaloidal parametrization for detecting critical two-view configurations

Supplementary Material

In this document, we provide additional details concerning the mathematical background of the generation of homaloidal nets through linear forms (Sec. A), the closest competitor of our method (Sec. B), the synthetic setup used to simulate critical configurations (Sec. C) and further visualizations on a rendered and real scene (Sec. D).

A. On the generation of homaloidal nets through linear forms

We provide here an explicit example where we follow step by step the proposed formulation of Sec. 4.1 and 4.2.

First of all, we notice that in formula (5) L_i and M_i are linear forms $L_i, M_i : \mathbb{R}^3 \to \mathbb{R}$. Evaluating them at (x_0, x_1, x_2) and collecting the values into vectors $L = [L_0, L_1, L_2]$ and $M = [M_0, M_1, M_2]$, then, due to Equation (9), $\mathbf{y} = [y_0, y_1, y_2]$ can indeed be interpreted as proportional to the cross product $L \times M$. This gives an other interpretation of the link between Eq. (8) and (9).

A.1. Example

Consider the two pairs of projection matrices:

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{14}$$

$$Q_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{15}$$

It is immediate to see that they are not projectively equivalent. Following the procedure described in Sec. C, it can be verified that the critical quadric corresponding to these camera matrices has the following equation:

$$x_0 x_1 + x_2 x_3 = 0, \tag{16}$$

and the quadratic transformation ψ (see e.g. [6]) is given by:

$$\begin{cases} \rho y_0 = x_0 x_2 \\ \rho y_1 = x_1 x_2 \\ \rho y_2 = -x_0 x_1. \end{cases} \tag{17}$$

Hence the three base points are: $(1,0,0), (0,1,0), (0,0,1)$. In this case, it is possible to take the following linear forms:

$$\begin{pmatrix} L_0 & L_1 & L_2 \\ M_0 & M_1 & M_2 \end{pmatrix} = \begin{pmatrix} 0 & x_0 & x_2 \\ x_1 & 0 & x_2 \end{pmatrix}. \tag{18}$$

As proved in Prop. 3, this is not the only possible choice.

To conclude, we observe that the example described above is the simplest possible for two reasons: first, the three base points are the vertices of the coordinate system, and second, the three conics that generate the net are the degenerate conics consisting of pairs of lines passing through the three base points. In general, the base points of the quadratic transformation will be three arbitrary points that can be transformed into the vertices of the coordinate system by an homography in the coordinate plane (x_0, x_1, x_2). In addition, the conics that generate the net will not be the reducible ones, but other cases are possible; however, even in this case, from a purely theoretical point of view, one can search the degenerate conics in the net and use these to generate the net. This last operation corresponds to having performed a projective transformation of the plane (y_0, y_1, y_2).

B. Luong-Faugeras Algorithm

In Algorithm 2, we report the pesudocode detailing the method of Luong and Faugeras [19] for estimating the quadratic transformation ψ described in Sec. 3. The method first estimates a fundamental matrix F_P from 7 correspondences (Line 1), obtaining up to three solutions F_P^j with $j = 1, \dots, 3$. For each solution, the best alternative fundamental matrix F_Q^j is estimated by solving a linear system (Lines 3–4) and refined via geometric optimization (Line 5). The final pair F_P, F_Q is the one with minimum geometric transfer error (Line 7), which in turn defines a quadratic transformation (Line 8). See Sec. 3 for more details.

Algorithm 2 Luong-Faugeras fitting of ψ

Require: $n \geq 7$ corresponding points $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$
Ensure: A quadratic transformation ψ
1: Compute fundamental matrices F_P^j from the corresponding points $(\mathbf{x}_i, \mathbf{y}_i)$ using the 7-point algorithm (up to 3 possible solutions F_P^1, F_P^2, F_P^3)
2: **for** each F_P^j **do**
3: Construct the 7×9 matrix A such that the i-th row A_i is given by $A_i = \mathbf{x}_i^\top \otimes [\mathbf{y}_i]_\times [F_P^j \mathbf{x}_i]_\times$
4: Solve for F_Q^j such that $A\text{vec}(F_Q^j) = 0$
5: Refine F_Q^j by minimizing the error in Eq. (4)
6: **end for**
7: Choose F_P, F_Q corresponding to $\min e_j$.
8: $\psi(\mathbf{x}) = F_P \mathbf{x} \times F_Q \mathbf{x}$.

Algorithm 2 was selected as a competitor because, to the best of our knowledge, it is the only prior work that explicitly targets the same degeneracy detection problem. Al-

though older, it remains the most relevant baseline in terms of problem formulation. This also highlights that our work addresses a long-standing problem for which no alternative practical solution has been proposed so far.

C. Synthetic 3D Data Generation

We now provide further details on the procedure used for our synthetic experiments detailed in Sec. 5. See Fig. 9 for a visualization. Specifically, to generate $3D$ critical points and their $2D$ correspondences, we proceed as follows:

1. We generate a set of four random cameras, divided into two pairs (P_1, P_2) and (Q_1, Q_2), ensuring that they are not projectively equivalent.
2. For each pair of cameras, we compute the ground-truth fundamental matrices F_P and F_Q using closed-form formulas from [16].
3. Using Lemma 5.10 from [15], we compute the matrix A identifying the ruled quadric containing the camera centers of (P_1, P_2): $A = \frac{S^\intercal + S}{2}$, where $S = P_2^\intercal F_Q P_1$.
4. To pick a 3D point X lying on the surface, we proceed as described in Fig. 9:
   - We sample a random base point $\mathbf{m}_0$ in 3D space and a random direction $\mathbf{v}$ of unit length. A critical point, X, is computed as the intersection of a point on the line $X = \mathbf{m}_0 + t\mathbf{v}$ and the surface, $X^\intercal S X = 0$.
   - To add noise in 3D, the point is moved along the line $\mathbf{m}_0 + t\mathbf{v}$, by a distance θ, away from the surface.
5. We pick a set of 3D points as in Step 4, which guarantees that they are in a critical configuration (when $\theta = 0$).
6. The above 3D points are then projected onto 2D images via cameras P_1 and P_2, producing a set of correspondences in $2D$. We then precondition the correspondences to improve numerical stability as in [16].

In our simulations, we use the hyperboloid as a representative example, since all smooth critical surfaces induce equivalent critical configurations in two views. Degenerate quadrics correspond to homography-induced degeneracies, which are also correctly identified by our method (as shown in Fig. 7 from the main paper).

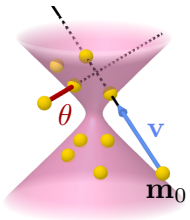

Figure 9. Data generation of critical points, lying on a ruled quadric surface, in 3D. The parameter θ is used to perturb the points away from the critical surface.

D. More Simulated and Real Degeneracies

We rendered the synthetic scene in Fig. 10 in BLENDER, consisting of a cylindrical surface observed by two cameras with centers lying on the cylinder. Using the known ground-truth fundamental matrix, we verified that: i) all true correspondences are correctly detected as critical with our method; ii) the estimated fundamental matrix is incorrect and unstable. These results further support our findings.

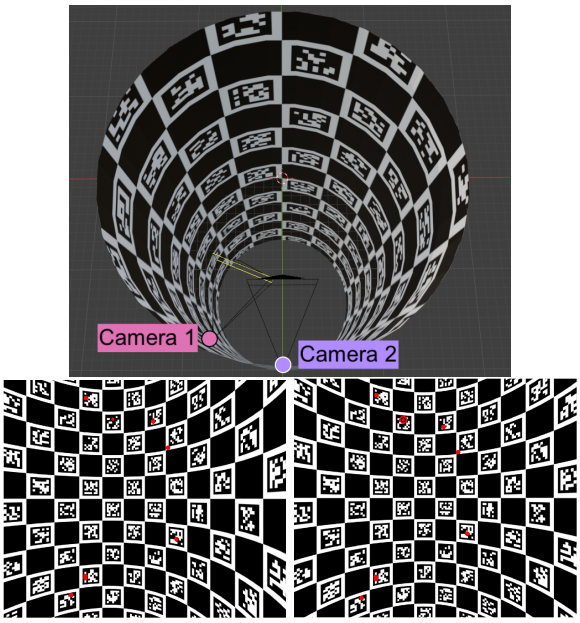

Figure 10. Two rendered views of a cylindrical surface result in unstable fundamental matrix estimation. We detect this configuration as critical.

Finally, we extend our analysis in Sec. 5.3 by providing further examples of degenerate point correspondences from pairs of images taken in the real square depicted in Fig. 8. These additional visualizations are shown in Fig. 11. We recall that $\text{err}(\mathbf{x_8}, \mathbf{y}_8)$ measures the 2D error under the quadratic transformation ψ of the 8-th correspondence (i.e., between $\psi(\mathbf{x}_8)$ and $\mathbf{y}_8$ expressed in normalized coordinates): $\text{err}(\mathbf{x_8}, \mathbf{y}_8) = ||\mathbf{y}_8 - \psi(\mathbf{x}_8)||$. A low error means that the quadratic transformation describes well the entire set of 8 correspondences, i.e., the configuration is critical. Similar to the case in Fig. 8, the circle ◯ marks the estimated correspondence $\psi(\mathbf{x}_8)$ obtained from the fitted quadratic transformation: it is clear that $\psi(\mathbf{x}_8)$ is always close to $\mathbf{y}_8$ with errors between 10^{-6} and 10^{-4} pixels, in the normalized space: the configurations are correctly identified as critical, providing further qualitative evidence of the usefulness of our degeneracy detection approach.

Figure 11. More qualitative results of our degeneracy test on a real-world critical configuration, depicting a square with a cylindrical layout captured from viewpoints near the surrounding facades. The dots • denote correspondences in the two images; one of them, $(\mathbf{x}_8, \mathbf{y}_8)$, is highlighted with a thick ∘ black border. The circle ◯ marks the estimated correspondence $\psi(\mathbf{x}_8)$ obtained from the fitted quadratic transformation: $\psi(\mathbf{x}_8)$ is actually close to $\mathbf{y}_8$ with a low error for all the pairs, i.e., the configuration is correctly identified as critical with our approach. *Zoom in for details*.