4 Formulation II: Camera From Ray Map

DLT / RQ Decomposition & Tokens

Deriving Parameters from Ray Map M

Given ray map $M \in \mathbb{R}^{H \times W \times 6}$ with origins $M_{:3}$ and directions $M_{3:}$:

1 Estimate Camera Center t_c

$$t_c = \frac{1}{H \cdot W} \sum_{h,w} M(h, w, :3)$$

2 Recover K, R via Homography

Canonical ray $d_I = p$ relates to camera ray d_{cam} via $H = KR$.

$$H^* = \arg \min_{||H||=1} \sum_{h,w} ||(Hp_{h,w}) \times M(h, w, 3:)||$$

Why cross product?
Minimizes angular error—enforces directional alignment between $H \cdot p$ and predicted rays.

Solved via DLT, then decompose H^* using RQ decomposition → (K, R).

Lightweight Camera Head D_C

Challenge: Pose-from-rays optimization is computationally expensive at inference.

Solution: A dedicated camera head operating on camera tokens directly predicts (f, q, t) parameters with **negligible overhead**—bypassing expensive DLT/RQ at test time.

Camera Conditioning Tokens

Camera information is injected via tokens prepended to each view, enabling both posed and unposed inputs.

`If pose known:`

$$c_i = E_c(f_i, q_i, t_i)$$

Encoded via MLP E_c from FOV, quaternion, translation.

`If pose unknown:`

Use a shared learnable token c_ℓ.

TOKEN INTEGRATION FLOW

1. Image Patches I_i ⊕ Camera Token c_i

↓

2. ViT Backbone
Joint Processing (All Layers)

↓

3. Geometry Features | Updated c_i