Algorithm 14.3: Inferring 3D world points (reconstruction)

Given J calibrated cameras in known positions (i.e. cameras with known $\boldsymbol{\Lambda}, \boldsymbol{\Omega}, \boldsymbol{\tau}$), viewing the same three-dimensional point $\mathbf{w}$ and knowing the corresponding projections in the images $\{\mathbf{x}_j\}_{j=1}^J$, establish the position of the point in the world.

As for the previous algorithms the final solution depends on a non-linear minimization of the reprojection error between $\mathbf{w}$ and the observed data $\mathbf{x}_j$,

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \left[\sum_{j=1}^{J} (\mathbf{x}_j - \mathbf{pinhole}[\mathbf{w}, \boldsymbol{\Lambda}_j, \boldsymbol{\Omega}_j, \boldsymbol{\tau}_j])^T (\mathbf{x}_j - \mathbf{pinhole}[\mathbf{w}, \boldsymbol{\Lambda}_j, \boldsymbol{\Omega}_j, \boldsymbol{\tau}_j]) \right]$$

The algorithm below finds a good approximate initial conditions for this minimization using a closed-form least-squares solution.

Algorithm 14.3: Inferring 3D world position

Input : Image points $\{\mathbf{x}_j\}_{j=1}^J$, camera parameters $\{\boldsymbol{\Lambda}_j, \boldsymbol{\Omega}_j, \boldsymbol{\tau}_j\}_{j=1}^J$
Output: 3D world point $\mathbf{w}$
begin
  for j=*1* **to** J **do**
    `// Convert to normalized camera coordinates`
    $\mathbf{x}'_j = \boldsymbol{\Lambda}_j^{-1}[x_j, y_j, 1]^T$
    `// Compute linear constraints`
    $a_{1j} = [\omega_{31j}x'_j - \boldsymbol{\omega}_{11j}, \omega_{32j}x'_j - \omega_{12j}, \omega_{33j}x'_j - \omega_{13j}]$
    $a_{2j} = [\omega_{31j}y'_j - \boldsymbol{\omega}_{11j}, \omega_{32j}y'_j - \omega_{12j}, \omega_{33j}y'_j - \omega_{13j}]$
    $b_j = [\tau_{xj} - \tau zjx'_j; \tau_{yj} - \tau_{zj}y'_j]$
  end
  `// Stack linear constraints`
  $\mathbf{A} = [a_{11}; a_{21}; a_{12}; a_{22}; \ldots a_{1J}; a_{2J}]$
  $\mathbf{b} = [b_1; b_2; \ldots b_J]$
  `// LS solution for parameters`
  $\mathbf{w} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$
  `// Refine parameters with non-linear optimization`
  $\hat{\mathbf{w}} = \operatorname{argmin}_{\mathbf{w}} \left[\sum_{j=1}^J (\mathbf{x}_j - \mathbf{pinhole}[\mathbf{w}, \boldsymbol{\Lambda}_j, \boldsymbol{\Omega}_j, \boldsymbol{\tau}_j])^T (\mathbf{x}_j - \mathbf{pinhole}[\mathbf{w}, \boldsymbol{\Lambda}_j, \boldsymbol{\Omega}_j, \boldsymbol{\tau}_j])\right]$
end