Calibrated Camera Resectioning Using P3P in RANSAC Scheme

A[E]4M33TDV—3D compute vision: labs.

Martin Matoušek

November, 2010*

Lecture Prerequisities: Three-Point Exterior Orientation Problem (P3P), Camera Resectioning.

1 Task Formulation

The task of camera resectioning is to estimate pose of a camera, given a set of n known 3-D points in space and its known projections in the image by that camera. I.e., every image point $\boldsymbol{u}_i$ corresponds to a spatial point $\boldsymbol{X}_i$,

$$\boldsymbol{u}_i \leftrightarrow \boldsymbol{X}_i \, \forall i = 1 \dots n \,. \tag{1}$$

This set of image-to-scene point correspondences is tentative, it can contain erroneous correspondences.

The camera internal calibration matrix $\mathbf{K}$ is known. The problem is to find camera absolute pose, i.e., its rotation $\mathbf{R}$ and translation $\boldsymbol{t} = -\mathbf{R}\boldsymbol{C}$ such that

$$\boldsymbol{u}_i \simeq \mathbf{P}\boldsymbol{X}_i \,, \forall i = 1 \dots n \,, \quad \text{where} \quad \mathbf{P} = \mathbf{K}[\mathbf{R} \mid \boldsymbol{t}] \,. \tag{2}$$

2 Resectioning in RANSAC Scheme

Since the input tentative correspondences can contain errors, a robust method is needed for camera pose estimation. Here, the RANSAC scheme is used. As usual, every RANSAC iteration in this task consists of two parts: the first is generation of a hypothesis, the second is verification of the hypothesis. These are repeated, according to the RANSAC stopping criterion (number of iterations), based on number of inliers and given probability. When computing the number of iterations, we recommend to use $-\log(1-p)$ in range $5-50$, but it can depend on the scene.

1. Check, if there are enough correspondences. If not, terminate with no solution found.
2. Initialise number of iterations and its allowed maximum,

$$N_{\text{iter}} \leftarrow 0 \,, \quad N_{\text{max}} \leftarrow \infty \,. \tag{3}$$

3. Initialise best hypothesis and support,.

$$\text{support}_{\text{best}} \leftarrow 0 \,, \quad \mathbf{R}_{\text{best}} \leftarrow [] \,, \quad \mathbf{t}_{\text{best}} \leftarrow [] \,. \tag{4}$$

4. Next iteration. Set $N_{\text{iter}} \leftarrow N_{\text{iter}} + 1$. If $N_{\text{iter}} \leq N_{\text{max}}$, continue, otherwise terminate with the best hypothesis found. There can be no solution also.

*Last revision: November 22, 2010

5. Generate hypotheses (Section 2.1).

6. For each hypothesis $\mathbf{R}$, $\boldsymbol{t}$ found:

   (a) Verify the hypothesis computing the set of inliers $\mathcal{I}_{\text{inl}}$ and support (Section 2.2).

   (b) If support $>$ support$_{\text{best}}$ then update the best hypothesis found,

$$\begin{aligned} \text{support}_{\text{best}} &\leftarrow \text{support}\,, & (5)\\ \mathbf{R}_{\text{best}} &\leftarrow \mathbf{R}\,, & (6)\\ \boldsymbol{t}_{\text{best}} &\leftarrow \boldsymbol{t}\,. & (7)\\ & & (8) \end{aligned}$$

   Also update RANSAC stopping criterion w.r.t. the number of inliers,

$$\epsilon = 1 - \frac{|\mathcal{I}_{\text{inl}}|}{n}\,, \quad N_{\text{max}} \leftarrow \begin{cases} 0 & \text{if } \epsilon = 0\,, \\ \frac{\log(1-p)}{\log(1-(1-\epsilon)^3)} & \text{otherwise}\,. \end{cases} \tag{9}$$

7. Continue by Step 4.

2.1 Generation of a hypothesis

1. Random sample of three point correspondences: randomly select $i_1, i_2, i_3 \in \langle 1, n \rangle$ such that they are unique.

2. Using $\{\mathbf{K}^{-1}\boldsymbol{u}_{i_1}, \mathbf{K}^{-1}\boldsymbol{u}_{i_2}, \mathbf{K}^{-1}\boldsymbol{u}_{i_3}\} \leftrightarrow \{\boldsymbol{X}_{i_1}, \boldsymbol{X}_{i_2}, \boldsymbol{X}_{i_3}\}$ estimate camera pose using P3P (code provided). The $\mathbf{R}$ and $\boldsymbol{t}$ are obtained. Note that P3P is intended for a calibrated case, so it works with **image points after the K is undone**. Several solutions (pairs $\mathbf{R}$, $\boldsymbol{t}$) can arise here, every one is later verified separately.

2.2 Verification of hypothesis

A hypothesis for camera pose, characterised by $\mathbf{R}$ and $\boldsymbol{t}$, is now verified against all correspondences.

1. Construct camera matrix $\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \boldsymbol{t}]$.

2. Select $\mathcal{I}_{\text{front}}$—the 3D points that are in front of the camera,

$$([\mathbf{R} \mid \boldsymbol{t}]\boldsymbol{X}_i)_3 > 0\,, \forall i \in \mathcal{I}_{\text{front}}\,. \tag{10}$$

3. Project the selected points by the camera and compute **Euclidean reprojection errors** (squared) w.r.t. original image points $\boldsymbol{u}_i$,

$$\hat{\boldsymbol{u}}_i \simeq \mathbf{P}\boldsymbol{X}_i\,, i \in \mathcal{I}_{\text{front}}\,, \tag{11}$$

$$\boldsymbol{e}_i = \begin{bmatrix} \frac{\hat{u}_i(1)}{\hat{u}_i(3)} \\ \frac{\hat{u}_i(2)}{\hat{u}_i(3)} \end{bmatrix} - \begin{bmatrix} \frac{u_i(1)}{u_i(3)} \\ \frac{u_i(2)}{u_i(3)} \end{bmatrix}\,, \tag{12}$$

$$e_i^2 = d^2(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i) = \boldsymbol{e}_i^\top \boldsymbol{e}_i\,. \tag{13}$$

   Note that the errors are measured in image domain, i.e., the $\mathbf{P}$ is full camera matrix including $\mathbf{K}$ and the error e_i is **in pixels**.

4. Compute set of inliers and support. For a given threshold e_{thr}, a set of inliers is

$$i \in \mathcal{I}_{\text{inl}} \iff i \in \mathcal{I}_{\text{front}} \wedge e_i^2 \leq e_{\text{thr}}^2 \,. \tag{14}$$

Two approaches can be used for computing the support. Simpler approach uses only the result of thresholding the error, thus every inlier contribute to support by 1,

$$\text{support} = \sum_{i \in \mathcal{I}_{\text{inl}}} 1 \;=\; |\mathcal{I}_{\text{inl}}| \,. \tag{15}$$

Better approach is to approximate ML estimator, where every inlier contribute as $1 - \frac{e_i^2}{e_{\text{thr}}^2}$,

$$\text{support} = \frac{1}{n} \sum_{i \in \mathcal{I}_{\text{inl}}} \left(1 - \frac{e_i^2}{e_{\text{thr}}^2}\right) . \tag{16}$$

Here the normalising term does not affect the algorithm, since n is constant, but this relative support can be compared across different cameras.