3D Computer Vision - Task 0-3 notes

Lab session materials for subjects B4M33TDV, BE4M33TDV, XP33VID

Martin Matoušek

October 2020

CENTER FOR MACHINE PERCEPTION

Revision: October 7, 2020

The aim is to find a model M that best suits the data.
$\{x_1, \ldots, x_N\}$ – the data set (points, correspondences, etc.)
N – number of data elements
n – minimum number of data elements needed to estimate the model
$\varepsilon_M(x_i)$ – error of a data element x_i with respect to the model M
parameters: threshold θ, probability p

Algorithm:

1. init: $k \leftarrow 0$, best-support $\leftarrow 0$
2. iteration: $k \leftarrow k + 1$
3. sample: randomly draw n data points
4. hypothesis: compute a model M_k from the sample
5. evaluate error $\varepsilon_{M_k}(x_i)$ of every data point in the set w.r.t. M_k
6. inliers: $\mathcal{I} = \left\{x_i \mid \varepsilon_{M_k}(x_i) < \theta\right\}$ – data with error smaller then threshold
7. support $\leftarrow N_{\mathcal{I}}$, $\quad N_{\mathcal{I}} = |\mathcal{I}|$ – number of inliers
8. update: if support $>$ best-support
   best-support $\leftarrow$ support
   $M^* \leftarrow M_k$
   $N_{\text{max}} = \frac{\log(1-p)}{\log(1-w^n)}$ (stopping criterion), where $w = \frac{N_{\mathcal{I}}}{N}$ (inlier ratio)
9. terminate with M^* if $k > N_{\text{max}}$, otherwise repeat from step 2

MLESAC

- Standard RANSAC uses 0-1 (box) support function

$$s_i = \begin{cases} 1 & \text{if } \varepsilon(x_i) \le \theta \\ 0 & \text{otherwise} \end{cases}$$

$$\text{support} = \sum_i s_i$$

- MLESAC – modification of support for maximum likelihood error on inliers

$$s_i = \begin{cases} 1 - \frac{\varepsilon(x_i)^2}{\theta^2} & \text{if } \varepsilon(x_i) \le \theta \\ 0 & \text{otherwise} \end{cases}$$

Point-line error (orthogonal distance)

$$\boldsymbol{l}^\top = \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$\varepsilon_{\boldsymbol{l}}(\boldsymbol{x}) = \frac{\boldsymbol{l}^\top \boldsymbol{x}}{w\,\sqrt{l_1^2 + l_2^2}}$$

normalized parameterisation

$$\boldsymbol{l}^\top = \begin{bmatrix} \boldsymbol{n}^\top & d \end{bmatrix} \quad |\boldsymbol{n}| = 1$$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{u} \\ 1 \end{bmatrix}$$

$$\varepsilon_{\boldsymbol{l}}(\boldsymbol{u}) = \boldsymbol{n}^\top \boldsymbol{u} + d$$

Minimize sum of squared point-line distances w.r.t. line parameters: $\boldsymbol{n}^*, d^* = \arg\min C(\boldsymbol{n}, d)$

$$C(\boldsymbol{n}, d) = \sum_{i=1}^{N} (\varepsilon_{\boldsymbol{l}}(\boldsymbol{u}_i))^2 = \sum_{i=1}^{N} \left(\boldsymbol{n}^\top \boldsymbol{u}_i + d\right)^2 = \sum_{i=1}^{N} (\boldsymbol{n}^\top \boldsymbol{u}_i)^2 + 2d\boldsymbol{n}^\top \sum_{i=1}^{N} \boldsymbol{u}_i + Nd^2$$

1. find d – zero first derivative gives extrema

$$\frac{\partial C}{\partial d} = 2\boldsymbol{n}^\top \sum_{i=1}^{N} \boldsymbol{u}_i + 2Nd = 0 \rightarrow \boxed{d = -\boldsymbol{n}^\top \boldsymbol{\mu}_u} \quad \text{where} \quad \boldsymbol{\mu}_u = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{u}_i \quad \text{(centroid)}$$

2. let $\boldsymbol{u}'_i = \boldsymbol{u}_i - \boldsymbol{\mu}_u$, substitute d and $\boldsymbol{u}_i = \boldsymbol{u}'_i + \boldsymbol{\mu}_u$ to $C(\boldsymbol{n}, d)$

$$\begin{aligned} C(\boldsymbol{n}, d) &= \sum \left(\boldsymbol{n}^\top (\boldsymbol{u}'_i + \boldsymbol{\mu}_u) - \boldsymbol{n}^\top \mu_u\right)^2 = \sum \left(\boldsymbol{n}^\top \boldsymbol{u}'_i\right)^2 = \boldsymbol{n}^\top \left(\sum \boldsymbol{u}'_i \boldsymbol{u}'^\top_i\right) \boldsymbol{n} = \\ &= \boldsymbol{n}^\top \mathbf{A}^\top \mathbf{A} \boldsymbol{n} \quad \text{where} \quad \boxed{\mathbf{A}^\top = [\ldots, (\boldsymbol{u}_i - \boldsymbol{\mu}_u), \ldots]} \quad \text{(stacked points)} \end{aligned}$$

Let $\boxed{\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top}$ (singular value decomposition), then $C(\boldsymbol{n}, d) = \boldsymbol{n}^\top \mathbf{V}\mathbf{D}^2\mathbf{V}^\top \boldsymbol{n}$, and minimum is for $\boldsymbol{n}^\top \mathbf{V} = [0, \ldots, 0, 1]$ (must be unit vector, so select the smallest singular value)

$$\mathbf{V} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 \end{bmatrix} \quad \rightarrow \quad \boxed{\boldsymbol{n} = \boldsymbol{v}_2}$$

Alternative: instead of computing s.v.d. of $N \times 2$ matrix $\mathbf{A}$, compute eigen values/vectors of 2×2 matrix $\mathbf{A}^\top \mathbf{A}$. The solution $\boldsymbol{n}$ is then eigenvector corresponding to the smallest eigenvalue.