Exercise 3D Machine Vision

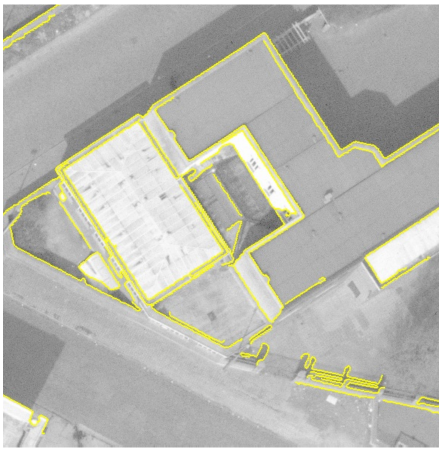
Prof. Dr.-Ing. Volker Willert

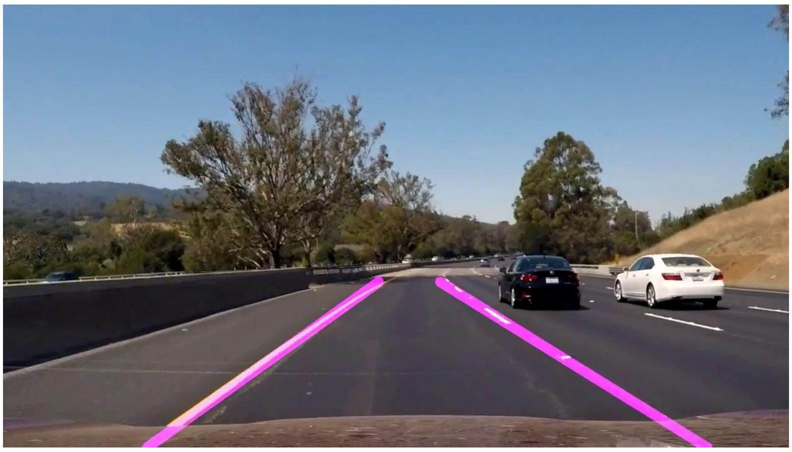
Sheet 3

In many computer vision and point-cloud processing tasks, a lot of sub-problems can be solved with regression. For example

- the determination of intrinsic camera parameters based on 2D-3D point correspondences,
- fitting a linearly parameterized 2D curve (e.g. straight line or parabola) into a set of image point coordinates, or
- the calculation of the camera orientation based on more than three world points, etc...

Especially the fitting of contours is formulated as a least squares problem. Given are 2D coordinates along a contour. We are looking for the model parameters of the contour, e.g. of a straight line. Such calculations are used in the field of automated mapping or visual driver assistance.





The basic problem of supervised model fitting is as follows:
Defining N pairs of input data and targets $(\mathbf{x}_i, t_i)$, $i = 1, ..., N$ and $k < N$ functions f_1 to f_k. What we are looking for is a linear combination $f(\mathbf{x}) = \sum_{j=1}^{k} w_j f_j(\mathbf{x})$ of these f_j such that the sum of squared deviations of f at the points $\mathbf{x}_i$ to t_i becomes minimal:

$$\text{argmin}_{\mathbf{w}} \sum_{i=1}^{N} (f(\mathbf{x}_i) - t_i)^2 = \text{argmin}_{\mathbf{w}} \sum_{i=1}^{N} (\sum_{j=1}^{k} w_j f_j(\mathbf{x_i}) - t_i)^2$$

This problem is also called *Gauß's method of least squares* or *regression*.

Task 3.1: Line Fitting

The following $N = 3$ image point coordinates: $(x_1, y_1) = (1, 1)$, $(x_2, y_2) = (2, 1)$, $(x_3, y_3) = (3, 2)$ are given.

- Specify all k functions f_j, stack them to the measurement matrix and determine the pseudoinverse.
- Calculate the optimal parameters $\mathbf{w}^\star$ of the straight line using the pseudoinverse.
- Rate the quality of your line model via the correlation coefficient.

Task 3.2: Depth value interpolation via 2nd order polynomial fit

Given four gray values at four neighboring pixels arranged in a square in a depth image:
$Z(100, 50) = 6$, $Z(101, 50) = 10$, $Z(100, 51) = 4$ and $Z(101, 51) = 1$.
Approximate the depth values Z_i of this 2×2-size image section with a second order polynomial
$Z_i \approx w_1 x_{1i}^2 + w_2 x_{1i} x_{2i} + w_3 x_{2i}^2$.

Task 3.3: The RANSAC Algorithm

Given are four image point coordinates $\mathbf{x}_1 = (1, 1)$, $\mathbf{x}_2 = (2, 1)$, $\mathbf{x}_3 = (3, 2)$ and $\mathbf{x}_4 = (1, 3)$, where three points are near a straight edge and one is not (outlier). Determine these three points and the straight line equation that best approximates the edge using the RANSAC algorithm. Proceed as follows:

- Draw the four points in a coordinate system.
- RANSAC Step 1: Randomly select as many points from the data points as needed to calculate the parameters of the model. This is done with the expectation that this set is free of outliers. (For "exercise" purposes, choose $\mathbf{x}_1$ and $\mathbf{x}_4$ in the first iteration step, and $\mathbf{x}_2$ and $\mathbf{x}_3$ in the second.)
- RANSAC Step 2: Determine the model parameters (straight line parameters) with the chosen points.
- RANSAC Step 3: Determine the subset of measured values whose distance to the model curve is smaller than a certain limit g (for this example $g = 0.5$ is chosen). This subset is called *consensus set*. If it contains a certain minimum number of values, a good model has probably been found and the consensus set is saved.
- Repeat steps $1 - 3$ several times.

The RANSAC algorithm depends mainly on three parameters:

1. the maximum distance of a data point from the model up to which a point is not considered a gross error;
2. the number of iterations; and
3. the minimum size of the consensus set, i.e., the minimum number of points consistent with the model.

- What happens if you choose the limit $g = 0.1$?
- What happens if you choose the limit $g = 2$?
- What follows from this for the choice of the limit value?

The minimum number n of required repetitions depends only on the relative outlier fraction ϵ, the number of model parameters s as well as the probability of the occurrence of at least one outlier-free subset with n repetitions $p_n(outlier = 0)$, but not on the total number of measured values. It can be calculated as follows:

$$n = \frac{\ln(1 - p_n(outlier = 0))}{\ln(1 - (1 - \epsilon)^s)} .$$

Please calculate the minimum number n of required repetitions for fitting a line into a point cloud of 4 points including one outlier. The probability of selecting at least one outlier-free subset from all data points is is set to $p_n(outlier = 0) = 0.99$.