AS5

A20423189

1

(a) An outlier is a data point that diverges from an overall pattern in a sample. It influences the model fitting estimation and leads to not fitting the right model. Hence, we need to use algorithms to detect outliers and discard such points before model fitting.

(b) Objective function of robust estimation:

$$E(\theta) = \sum_{n=1}^{n} \rho_\sigma(d(x_i, \theta))$$

In robust estimation: $\rho_\sigma(x) = \frac{x^2}{x^2 + \sigma^2}$

Differences:
In standard least square objective function, Outliers will have higher value and influence the model more.

$$\rho_\sigma(x) = x^2$$

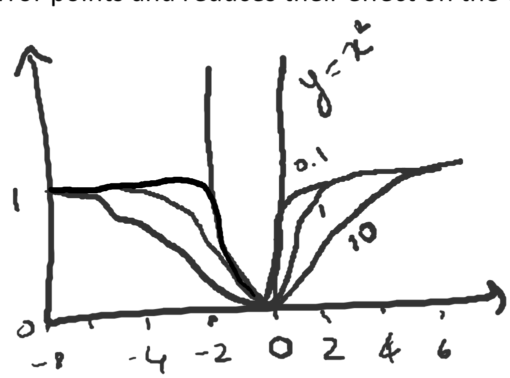
However, in robust estimate objective function will lower the influence of the outliers.

$$\rho_\sigma(x) = \frac{x^2}{x^2+\sigma^2}$$

(c) German-Mcclure estimator.

$$\rho_\sigma(x) = \frac{x^2}{x^2 + \sigma^2}$$

Advantages:
It will lower the influence of the outlier.
Captures the error points and reduces their effect on the objective function.



It has an upper bound.
If 6 is smaller, smaller valley, the model becomes more selective to outliers.

(d) Principle of RANSAC algorithm:
Repeat k times:
- Draw n points uniformly at random with replacement.
- Fit a model to points.
- Find inliers in entire set with distance less than t.
- Recompute model (if at least d inliers)
- Update parameters (k, t).

Number of points drawn at each attempt should be small in a hope that atleast one set will not have any outliers.

(e) Parameters:
N = # points drawn at each evaluation.
D = min # points needed to estimate model
K = # trails
t = distance threshold to identify inliers.

w = $\frac{number\ of\ inliers}{number\ of\ points}$

(1-p) = $(1 - w^n)^k$ => log(1 – p) => k log(1-w^n) => k $\frac{\log(1-p)}{\log(1-w^n)}$

(f) Objective of Image Segmentation:
Separating foreground from background.
2 Approaches of segmentation.
- Agglomerative(merge)
  Start with each pixel in a separate cluster. Merge cluster with small distance. Repeat while cluster are not satisfactory.
- Divisive(split)
  Start with all pixels in one cluster. Split cluster to produce large distance between them. Repeat while clusters are not satisfactory.

(g) k-Means:
- Select k
- Select initial guess of means:m_1, m_2, m_3, m_4.
- Take each pixel, choose closest cluster center and label. Recompute m_j
- Repeat until m_j change.

Mixture of Gaussians:

Instead of using $d = || f_i - m_j ||^2$ as the evaluation of distance it uses $d = (f_i - m_j)^T \sum_j (f_i - m_j)$

d = Mahalanobis distance

m_j = Mean of clusters.

j = covariance of cluster.

(h) Mean shift

$$m_j = \frac{\sum_{i \in S_j} w(f_i - m_j) f_i}{\sum_{i \in S_j} w(f_i - m_j)}$$

$$w(f_i - m_j) = \exp(-c \|f_i - m_j\|)$$

Similar to k-means. The mean computation is replaced with a weighted sum based on distance from center.

2

(a) Forward projection: Given world points P and translation matrix M, compute image point p.
Calibration: Given some points in real world with co-ordinates, compute some points in image with co-ordinates such as focal length, size of pixel, external translation and rotation.
Reconstruction: Given the image points p, compute the world points P.

Forward projection is the easiest and reconstruction is most difficult.

(b) Necessary input for camera calibration:
Corresponding points.
Image points in pixel and world points in meters.

(c) Steps:
1.Find projection matrix M

$$\underbrace{\begin{bmatrix} P_i^T & 0 & -x_i P_i^T \\ 0 & P_i^T & -y_i P_i^T \\ & \vdots & \\ P_m^T & 0 & -x_m P_m^T \\ 0 & P_m^T & -y_m P_m^T \end{bmatrix}}_{2m \times 12} \underbrace{\begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}}_{12 \times 1} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{2m \times 1}$$

The is a homogenous equation. The projection matrix can be estimated using SVD.
$A = u\, d\, v^t$
Solution is the column of v belonging to zero singular value.

2.Find parameters from M like K^*, R^* and T^*

$$\hat{M} = \begin{bmatrix} - & \hat{m}_1^T & - \\ - & \hat{m}_2^T & - \\ - & \hat{m}_3^T & - \end{bmatrix}$$

$$M = K^* [R^* | T^*] = \rho M^* = \rho \left[\begin{array}{c|c} - a_1^T - & \\ - a_2^T - & b \\ - a_3^T - & \end{array} \right]$$

$$|\rho| = 1/|a_3|$$

$$u_0 = |\rho|^2 a_1 . a_3$$

$$v_0 = |\rho|^2 a_2 a_3$$

$$a_v = \sqrt{|\rho|^2 a_2 . a_2 - v_0^2}$$

$$s = |\rho|^4 / a_v (a_1 \times a_3)(a_2 \times a_3)$$

$$a_v = \sqrt{|\rho|^2 a_1 . a_1 - s^2 - u_0^2}$$

$$K^* = \begin{bmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\epsilon = \operatorname{sgn}(b_3)$$

$$T^* = \epsilon |\rho| (K^*)^{-1} b$$

$$r_3 = \epsilon |\rho| a_3$$

$$r_1 = |\rho|^2 / \alpha_v \, a_2 \times a_3$$

$$r_2 = r_3 \times r_1$$

$$R^* = [r_1^T \quad r_2^T \quad r_3^T]^T$$

(d) $\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 1, 2, 3, 4 \\ 1, 0, 3, 4 \\ 1, 1, 1, 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}$

$= \begin{bmatrix} 18 \\ 14 \\ 7 \end{bmatrix}$

$\begin{bmatrix} x \\ y \end{bmatrix}$ 2D points is $\begin{bmatrix} 18/7 \\ 2 \end{bmatrix}$

(e) $\begin{bmatrix} 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & -100 & -200 & -300 & -100 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & -200 & -400 & -600 & -200 \\ & & & & & & & - & & & & \\ & & & & & & & - & & & & \\ & & & & & & & - & & & & \end{bmatrix} \begin{bmatrix} m_1 \\ - \\ - \\ - \\ - \\ m_2 \end{bmatrix} = \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \\ 0 \end{bmatrix}$

(f) Need at least 6 points pairs to get matrix M. i.e. For each point 2 equation with 12 unknows hence needs at least 6 points.

(g) $M = K^*[R^* | T^*] = \rho \hat{m}$

Since there are different ways of breaking and only one correct way we use the properties of matrix ie R^* is orthogonal (its a rotational matrix)

$K^*[R^* | T^*] = \rho \hat{m}$

$[K^* R^* | K^* T^*] = \rho \hat{m}$

$$K^* R^* = \begin{bmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} - & r_1^T & - \\ - & r_2^T & - \\ - & r_3^T & - \end{bmatrix} = \begin{bmatrix} \alpha_u r_1^T + s r_2^T + u_0 r_3^T \\ \alpha_v r_2^T + v_0 r_2^T \\ r_3^T \end{bmatrix}$$

$$\left[\begin{array}{c|c} \alpha_u r_1^T + s r_2^T + u_0 r_3^T & K^* T^* \\ \alpha_v r_2^T + v_0 r_3^T & \\ + r_3^T & \end{array} \right] = \rho \hat{M}$$

$$\rho \left[\begin{array}{c|c} - \; a_1^T \; - & b \\ - \; a_2^T \; - & \\ - \; a_3^T \; - & \end{array} \right] = \rho \hat{M}$$

$$\alpha_u r_1^T + s r_2^T + u_0 r_3^T = \rho a_1^T$$

$$\alpha_v r_2^T + v_0 r_3^T = \rho a_2^T$$

$$r_3^T = \rho a_3^T$$

$$K^* T^* = \rho b$$

$r_1 . r_2 = 0$	$r_1 \times r_2 = r_3$	$r_1 . r_1 = 1$
$r_2 . r_3 = 0$	$r_2 \times r_3 = r_1$	$r_2 . r_2 = 1$
$r_3 . r_1 = 0$	$r_3 \times r_1 = r_2$	$r_3 . r_3 = 1$

(h) Given the correspondence points (3D-2D) and the estimated projection matrix

$$M = \begin{bmatrix} - & m_1^T & - \\ - & m_2^T & - \\ - & m_3^T & - \end{bmatrix}$$

$$E = \frac{1}{m} \sum_i \left(\left| x_i - \frac{m_1^T P_i}{m_3^T P_i} \right| \right)^2 + \left(\left| y_i - \frac{m_2^T P_i}{m_3^T P_i} \right| \right)^2$$

distance between know & predited positions

(i) In order to perform planar camera calibration, we need to show the view to the camera for more than one time, minimum of three times to calibrate for planar.

- Estimate 2D homography (Projection Map) between calibration plane and image(for several images.
- Estimate intrinsic parameter
- Compute extrinsic parameters for view of interest.
  Planar solve 2DH points, Non-planar solve 3DH points.

In non-planar calibration, it uses 3D image as calibration target. So, we need to find the pixel of all the corners and based on these values camera calibration is done. While in planar, a single plane picture is used with different views.

(j) Difference between homography and projection matrix.
Homography : Projects a 3DH world point to a 2DH image point.
Projection Matrix: Projects a 2DH world point to a 2DH image point.

homography - 3x3 matrix
projection matrix - 3x4 matrix
homography - z axis is assumed to 0
projection matrix - z axis is considered.

$p_i = M P_i$

$M = K^*[r_1\ r_2\ r_3\ T^*]$ - projection m

$M = K^*[r_1\ r_2\ T^*]$ - homography
assume $z = 0$