Hw 4

1. (a) Outliers is noise points that is distant from other points. It's influence the model fitting value estimates. The outlier do not fit the right model, we need use algorithm to detect outlier and better the model.

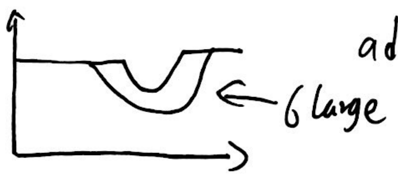
(b) $E(\theta) = \sum_{i=1}^{n} \rho_\sigma(d(X_i;\theta))$

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

In the standard least squares objective function, outliers will have higher value and influence model more. However, in robust estimate $\rho_\sigma(x) = \frac{x^2}{x^2+\sigma^2}$ will lower the influence of the outlier.

(c) German-McClure function: $\rho_\sigma(x) = \frac{x^2}{x^2+\sigma^2}$

It will lower the influence of outlier. if σ is bigger, the open range will bigger. if σ is smaller, the model become more selective of outlier.



advantages: 1). limit the effect of outlier.
2). capture the error.

(d) principle: Try k times, choice distance to get best model.

1). repeat k times:
- draw n points uniformally at random replacement
- Fit model to point
- find all inliers.
- if there d inliers. recompute model

2). choice best solution

the number of points drawn at each attempt should be small. Because, we only need the minimum points to fit the model, if we drawn too much points. it will contain more outliers.

(e) n = number points to draw,

d = minimum number points needed

k = number of trials

t = distance to identify inliners.

$$w = \frac{\text{\# Number of inliners}}{\text{Number of points}}$$

$$(1-p) = (1-w^n)^k \Rightarrow \log(1-p) = k\log(1-w^n) \Rightarrow$$

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

(f) segmentation: separate foreground from background.

Merge approach: Start with each pixel in separate cluster, iteratively merge clusters.

split approach: start with all pixels in one cluster, iteratively split clusters.

(g) k-means: - select k

- start with initial guess of k-means
- repeat: take each pixels, chose closet cluster center and ~~[illegible]~~ recompute m_j.
- stop until m_j do not change.

Mixture of Gaussians segmentation algorithm:

Instead of using $d = \| f_i - m_j \|^2$ as the evaluation of distance, It uses $d = (f_i - m_j)^T \Sigma_j^{-1} (f_i - m_j)$ (Σ_j is the covariance of sample.) It use normal distribution to weight the distance and reduce wrong labeling.

(h). Mean Shift: similar to k means, instead of m_j in k-means, use give a weighted from sample to the mean. It closer to the mean, more weighted or effect to the mean, mean shift find cluster centers as peaks of histgram.

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

2. (a) 1). forward projection: given world point P and translation matrix M, compute image point p

2). Calibration: given some points in real world with coordinates, compute some points in image with coordinates.

3). reconstruction: given image point p, compute world point P

forward projection is easiest, reconstruction is most difficult

(b) k^* : f_i (focal length), u_0, v_0 (optimal center)

s (scale) k_u, k_v (skew)

R^* T^* : R (Rotation), T (translation)

(c) step 1 : estimate projection Matrix M

step 2 : find parameter (k^*, R^*, T^*) given M.

(d) $P_i = (1, 2, 3, 1)$ in homogeneous coordinates.

$$P_i = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 18 \\ 14 \\ 7 \end{bmatrix}$$

$$P_i = \begin{bmatrix} \frac{18}{7} \\ 2 \end{bmatrix}$$ in 2D image coordinate.

(e)

$$\begin{bmatrix} 1\,2\,3\,1 & 0\,0\,0\,0 & -100 & -200 & -300 & -100 \\ 0\,0\,0\,0 & 1\,2\,3\,1 & -200 & -400 & -600 & -200 \\ & & \vdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ \vdots \\ \vdots \\ m_i \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

(f) need at least 6 points to get ~~1 eqs~~ matrix M

because we need to solve 11 unknown, in order to solve it, we need 6 points to get 12 equations.

(g) Break M into R^*, T^*, k^*

$$M = \rho \hat{M} \Rightarrow [k^*R^* \mid k^*T^*] = \rho \hat{M}$$

$$\hat{M} = \left[\begin{array}{c|c} \cdots a_1^T \cdots & \\ \cdots a_2^T \cdots & b \\ \cdots a_3^T \cdots & \end{array}\right] \qquad k^*R^* = \rho \begin{bmatrix} \cdots a_1^T \cdots \\ \cdots a_2^T \cdots \\ \cdots a_3^T \cdots \end{bmatrix}$$

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

(h) $$E(k^*, R^*, T^*) = \sum_{i=1}^{n} \left(x_i - \frac{M_1^T P_i}{M_3^T P_i}\right)^2 + \left(y_i - \frac{M_2^T P_i}{M_3^T P_i}\right)^2$$

(i) planar calibration steps:

1) estimate 2D homogenphy between calibration target and image

2) estimate intrinsic paremeters from serval views. (no change)

3) compute extrinsic paremeters for any views.

planar solve 2DH points, Non-planar solve 3DH points.

(j) 2DH: 3DH:

$$p_i = k^*[R^* \mid T^*] P_i$$

2DH:

$$\begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = \underbrace{k^*[r_1, r_2, T^*]}_{M^* \text{ homography}} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

$$P_i = \begin{bmatrix} x_i \\ y_i \\ 0 \\ 1 \end{bmatrix}$$

assume Z coordinate is zero.